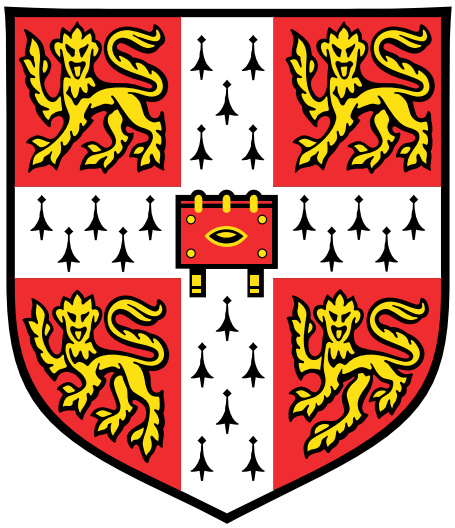
Statistically robust methods for the integration and analysis of X-ray diffraction data from pixel array detectors



James Michael Parkhurst
Laboratory of Molecular Biology
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

Wolfson College

August 2019

Declaration

I hereby declare that this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Acknowledgements and specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Acknowledgements and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Acknowledgements and specified in the text. This dissertation contains fewer than 60,000 words excluding appendices, bibliography, footnotes, tables and equations.

Abstract

New challenges in structural biology are driving the development of new technology, such as photon counting pixel array detectors, and new modes of data collection, such as serial synchrotron crystallography (SSX), in macromolecular crystallography at synchrotrons. This in turn is creating the need for better algorithms and software to extract the maximum amount of information from the diffraction data. The aim of this project is to develop statistically robust methods for the integration and analysis of X-ray diffraction data to address these challenges.

A method for estimating the background under each reflection during integration that is robust in the presence of pixel outliers is presented. This uses a generalised linear model (GLM) approach that is more appropriate for use with Poisson distributed data than traditional approaches to pixel outlier handling in integration programs. The algorithm is most applicable to data with a very low background level where assumptions of a normal distribution are no longer valid as an approximation to the Poisson distribution.

A second algorithm for modelling the background for each Bragg reflection in a series of X-ray diffraction images containing Debye-Scherrer diffraction from ice in the sample is also presented. This method involves the use of a global background model which is generated from the complete X-ray diffraction dataset. Fitting of this model to the background pixels is then done for each reflection independently.

Finally, a model for the observed reflection profiles is described for the purpose of improving the refinement of the crystal unit cell and orientation for still image diffraction data collected at synchrotrons. This model consists of two components: a Normal distribution is used to describe the distribution of wavelengths and a Multivariate Normal distribution (MVN) is used to describe the distribution of reciprocal lattice vectors for each reflection; this allows non-isotropic spot shapes to be easily described. The parameters of the model are estimated from the data *via* a simple maximum likelihood algorithm. The algorithms are incorporated into the *DIALS* integration package.

Dedicated to my grandfather

Acknowledgements

I would like to acknowledge the fact that I have been incredibly lucky. When I started at Diamond, I had the good fortune of joining Gwyndaf Evans' group. From the beginning, Gwyndaf encouraged me to apply to study for a PhD alongside my work at Diamond. During that time, I had the additional good fortune of meeting Garib Murshudov who then gave me the opportunity to study with him at the LMB. I could not have asked for two better supervisors and I would like to thank them both for the tremendous amount of guidance and support they have given me over the last few years.

At Cambridge, I would like to thank all of the many people with whom I have discussed my work; particularly Andrew Leslie, Phil Evans, Harry Powell, Paul Emsley, Robert Nicholls and Takanori Nakane who have given me help and advice about all matters pertaining to X-ray data processing. I am also immensely grateful to have met Andrea Thorn whilst at the LMB, with whom I started a collaboration that resulted in two journal articles. I would also like to thank my Cambridge supervisor, Randy Read, and my second LMB supervisor, Roger Williams, for their help and support throughout the project.

At Diamond, I would like to thank everyone in the DIALS team. In particular, Graeme Winter, David Waterman, Richard Gildea and Luis Fuentes Montero who were there from the beginning of the project and with whom I have discussed every aspect of my work. I would also like to express my gratitude to Melanie Vollmar, whose expertise in every aspect of crystallographic structure solution has been invaluable to me and with whom I have co-supervised four summer students at Diamond. I would also like to acknowledge Markus Gerstel, Nick Devenish, James Beilsten-Edmands and Ben Williams who have more recently joined the DIALS team.

I would like to thank CCP4 for not only providing me with funding for my PhD but also for providing me with so many wonderful opportunities to lecture at workshops around the world. I would like to thank everyone with whom I have worked or from whom I have received advice over the years. In no particular order: Charles Ballard, Andrei Lebedev, Eugene Krissnel, Ronan Keegan, Helen Ginn, Nick Sauter, Aaron Brewster, Danny Axford, John Beale, Ivo Tews, Rachel Bolton and many others. Special thanks go to Randy Read, Melanie Vollmar, David Waterman,

Danny Axford, John Beale and Sai Liu who were kind enough to read and give me feedback on some or all of my thesis. I would also like to thank my examiners Sjors Scheres and Kay Diederichs for providing me with excellent feedback and some final suggestions for improving the manuscript. Finally, I would like to thank my partner Andrea Tartakowsky for her constant help, support and encouragement.

Development of the *DIALS* software

The *DIALS* project is a collaborative effort with many contributors; however, the development has been structured such that individual components have a single main software developer. In order to be clear about my contribution to the *DIALS* project and the work that is described in Chapter 2, I will acknowledge the main authors of the core *DIALS* programs here.

I implemented many of the low-level software features in *DIALS*. I was responsible for designing the core data models used throughout the framework, I implemented the file handling and metadata storage used by all the *DIALS* programs, and I contributed C++ code and expertise throughout the project. As such, I have made a contribution to most parts of the software. For image metadata interpretation, handling of image formats was originally implemented by Graeme Winter; I used this as a basis for implementing a software library (*dxtbx*) for use in *DIALS*. For spot finding, I implemented all the software and algorithms. The indexing program was written by Richard Gildea. The centroid refinement program was written by David Waterman. I implemented all the software and algorithms for the profile refinement. For the integration, I implemented all the software and algorithms. The scaling program was written by James Beilsten-Edmands. Whilst Chapter 2 aims to describe data processing in general and the implementation within *DIALS* specifically, more detail is provided here for those parts of the data processing of which I was the major contributor.

Publications

I have authored or co-authored the following publications during the course of my PhD.

Parkhurst, J. M., A. S. Brewster, L. Fuentes-Montero, D. G. Waterman, J. Hattne, A. W. Ashton, N. Echols, G. Evans, N. K. Sauter, and G. Winter (2014). "DXTBX: the diffraction experiment toolbox". In: *Journal of Applied Crystallography* 47.4, pp. 1459–1465. DOI: 10.1107/s1600576714011996.

Parkhurst, J. M., G. Winter, D. G. Waterman, L. Fuentes-Montero, R. J. Gildea, G. N. Murshudov, and G. Evans (2016). "Robust background modelling in DIALS". In: *Journal of Applied Crystallography* 49.6, pp. 1912–1921. DOI: 10.1107/s1600576716013595.

Parkhurst, J. M., A. R. S. Thorn, M. Vollmar, G. Winter, D. G. Waterman, L. Fuentes-Montero, R. J. Gildea, G. N. Murshudov, and G. Evans (2017). "Background modelling of diffraction data in the presence of ice rings". In: *IUCrJ* 4.5, pp. 626–638. DOI: 10.1107/s2052252517010259.

Thorn, A. R. S., J. M. Parkhurst, P. Emsley, R. A. Nicholls, M. Vollmar, G. Evans, and G. N. Murshudov (2017). "AUSPEX: a graphical tool for X-ray diffraction data analysis". In: *Acta Crystallographica Section D* 73.9, pp. 729–737. DOI: 10.1107/s205979831700969x.

Winter, G., D. G. Waterman, J. M. Parkhurst, A. S. Brewster, R. J. Gildea, M. Gerstel, L. Fuentes-Montero, M. Vollmar, T. M. Michels-Clark, I. D. Young, N. K. Sauter, and G. Evans (2018). "DIALS: implementation and evaluation of a new integration package". In: *Acta Crystallographica Section D* 74.2, pp. 85–97. DOI: 10.1107/s2059798317017235.

Contents

1 Introduction **18**

1.1 Macromolecular X-ray Crystallography 18

1.1.1 Data collection at synchrotron facilities 19

1.1.2 Data reduction 20

1.1.3 Phasing 21

1.1.4 Refinement and model building 23

1.2 Integration software 24

1.2.1 Programs designed for rotation data 24

1.2.2 Programs designed for still data 25

1.2.3 The *DIALS* framework 26

1.3 Project motivation 26

1.3.1 New challenges in Crystallography 26

1.3.2 Micro crystallography 27

1.3.3 Pixel array detectors 27

1.3.4 Serial synchrotron crystallography 28

1.3.5 Project aims 29

2 Integration of X-ray diffraction data **31**

2.1 Introduction 31

2.2 Experimental geometry 34

2.2.1 Prediction of Bragg spots 35

2.2.2 Parallax correction 38

2.3 Interpretation of image metadata 38

2.3.1 Implementation in *DIALS* 40

2.4 Spot finding 44

2.5 Indexing 45

2.6 Refinement 49

2.6.1 Centroid refinement 49

2.6.2 Profile refinement 50

2.6.3 Post refinement 53

2.7 Integration 53

2.7.1 Background Estimation . . . 53
2.7.2 Summation . . . 54
2.7.3 Profile fitting . . . 56
2.7.4 Data correction . . . 62
2.8 Data reduction . . . 64

3 Robust background modelling using Generalised Linear Models 67
3.1 Introduction . . . 67
3.2 Algorithm . . . 70
3.2.1 Generalised linear models . . . 70
3.2.2 Robust estimation . . . 71
3.2.3 Robust GLM algorithm implementation in *DIALS* . . . 72
3.2.4 Background models . . . 74
3.2.5 Simplified algorithm for constant background model . . . 74
3.3 Analysis . . . 75
3.3.1 Experimental data . . . 75
3.3.2 Data analysis . . . 76
3.3.3 Background estimates . . . 77
3.3.4 Effects on data reduction . . . 82
3.4 Conclusion . . . 83

4 Background modelling in the presence of ice-rings 86
4.1 Introduction . . . 86
4.2 Algorithm . . . 90
4.2.1 Global background model . . . 90
4.2.2 Maximum likelihood fitting for each reflection . . . 92
4.2.3 Robust M-estimator for background scale factor . . . 93
4.3 Analysis . . . 94
4.3.1 Experimental data . . . 94
4.3.2 Refinement results . . . 95
4.3.3 Case studies . . . 95
4.3.4 Pixel statistics . . . 97
4.3.5 Intensity *versus* resolution . . . 97
4.3.6 Moments of E and R_{free} *versus* resolution . . . 100
4.3.7 The effect of noise on the intensity moments . . . 102
4.3.8 Application to data with no ice rings . . . 104
4.4 Conclusion . . . 105
4.4.1 Future improvements . . . 106

5 Profile modelling for serial synchrotron X-ray diffraction data 107
5.1 Introduction . . . 107
5.2 Algorithm . . . 111
5.2.1 Model . . . 111
5.2.2 General impact for δ-function wavelength model . . . 115
5.2.3 General impact for Normal wavelength model . . . 117
5.2.4 Parameter estimation . . . 122
5.2.5 Integration . . . 126
5.3 Analysis . . . 127
5.3.1 Experimental data . . . 127
5.3.2 Data analysis . . . 128
5.3.3 Analysis of crystal orientations . . . 130
5.3.4 Analysis of crystal unit cell parameters . . . 131
5.3.5 Analysis of positional residuals . . . 132
5.3.6 Analysis of predictions . . . 136
5.4 Conclusion . . . 138

6 Discussion 141
6.1 Conclusions . . . 141
6.1.1 Robust background modelling using Generalised Linear Models 142
6.1.2 Background modelling in the presence of ice-rings . . . 143
6.1.3 Profile modelling for serial synchrotron X-ray diffraction data 144
6.2 Future work . . . 146
6.3 Impact, deployment and use of the software . . . 147

A Appendix 150
A.1 GLM background algorithm usage in *DIALS* . . . 150
A.2 Ice ring background algorithm usage in *DIALS* . . . 150
A.3 Ice ring background data processing, reduction and refinement . . . 152
A.4 Block matrix inversion . . . 154
A.5 Product of the Ewald sphere and RLP distributions . . . 155
A.6 Derivatives of the RLP parametrisation . . . 156
A.7 Derivatives for δ-function wavelength model . . . 157
A.8 Derivatives for Normal wavelength model . . . 157
A.9 Still image data processing . . . 159

List of Figures

1.1 The percentage of structures in the PDB by experimental method. . . 19

2.1 The data flow within the *DIALS* framework. 33
2.2 The description of diffraction geometry for the rotation method. . . . 34
2.3 The Ewald sphere construction. 36
2.4 The CSPAD and PILATUS 12M-DLS detectors. 39
2.5 The image processing steps of the spot finding algorithm. 46
2.6 Multiple experiments. 51
2.7 The reflection shoebox. 55
2.8 Thickly and finely sliced data. 59
2.9 A reference profile used in profile fitting. 63

3.1 The average background level *versus* resolution. 77
3.2 Example reflections observed at 3Å. 78
3.3 Histogram of normalised differences in background. 80
3.4 Difference between the estimated and median background. 81
3.5 The cumulative distribution function for $|L|$ 84
3.6 The 4th acentric moment of E *versus* resolution 85

4.1 Intensity vs resolution for a dataset with strong ice rings. 88
4.2 Background estimation in the presence of ice rings. 89
4.3 The R_{free} after processing with different background models. 96
4.4 Diffraction images and average background. 98
4.5 The mean and dispersion images. 99
4.6 The intensity *versus* resolution from *AUSPEX*. 101
4.7 The 4th acentric moments of E *versus* resolution. 103

5.1 The SSX experimental setup on Diamond beamline I24. 109
5.2 The RLP distribution in reciprocal space. 112
5.3 The reflection specific coordinate system. 114
5.4 The predicted diffracted beam vector. 117
5.5 The product of the Ewald sphere and RLP distributions. 118
5.6 The distribution of Ewald spheres. 119

5.7 The Ewald sphere, RLP and product distributions. 121
5.8 Spot prediction. 127
5.9 An example of a 3Å spot for each dataset. 128
5.10 Analysis of crystal orientation. 131
5.11 Analysis of unit cell distortion. 133
5.12 The distribution of estimated unit cell parameters. 134
5.13 The distribution of positional residuals 137
5.14 The fraction of strong spots predicted. 139
5.15 The fraction of strong spots predicted *versus* resolution. 139
5.16 An example from each dataset for the spots predicted. 140

List of Tables

3.1 The percentage of reflections where all non-zero pixels were rejected. . 79
3.2 The twin fractions using each outlier handling algorithm. 82

4.1 A list of JCSG datasets with ice ring pathologies. 96

5.1 Data processing statistics. 130

A.1 Parameters to configure the background algorithm in *DIALS*. 151

Notation

Acronyms

AIC	Akaike Information Criteria.
DIALS	Diffraction Integration for Advanced Light Sources.
DUI	DIALS graphical User Interface.
EP	Experimental Phasing.
FFT	Fast Fourier Transform.
GLM	Generalised Linear Model.
GUI	Graphical User Interface.
IQR	Inter-Quartile Range.
JSON	JavaScript object notation.
MAD	Multiple-wavelength Anomalous Dispersion.
MIR	Multiple Isomorphous Replacement.
MIRAS	Multiple Isomorphous Replacement with Anomalous Scattering.
MR	Molecular Replacement.
MVN	Multivariate Normal.
MX	Macromolecular Crystallography.
ND	Neutron Diffraction.
NMR	Nuclear Magnetic Resonance.
PDB	Protein Data Bank.
RLP	Reciprocal Lattice Point.
SAD	Single-wavelength Anomalous Dispersion.
SFX	Serial Femtosecond Crystallography.
SIR	Single Isomorphous Replacement.
SIRAS	Single Isomorphous Replacement with Anomalous Scattering.
XFEL	X-ray Free Electron Laser.

General symbols

$\boldsymbol{a}^*$	A reciprocal lattice basis vector.
$\boldsymbol{B}$	The B factor.
$\boldsymbol{B}$	The crystal reciprocal space orthogonalisation matrix.
$\boldsymbol{b}^*$	A reciprocal lattice basis vector.

$\boldsymbol{c}^*$	A reciprocal lattice basis vector.
D	The index of dispersion.
$\boldsymbol{D}$	The detector projection matrix.
$\boldsymbol{d}$	The detector geometry matrix.
$\boldsymbol{e}_1$	A basis vector in the reflection specific coordinate system.
$\boldsymbol{e}_2$	A basis vector in the reflection specific coordinate system.
$\boldsymbol{e}_3$	A basis vector in the reflection specific coordinate system.
ϵ	A coordinate in the reflection specific coordinate system.
f_p	The polarisation fraction.
G	The gain of the detector.
g	The inverse scale factor.
$\boldsymbol{h}$	The reflection Miller index.
I	The observed intensity.
I_{bg}	The total summed background counts.
I_{corr}	The corrected intensity.
$< I_h >$	The observed intensity.
I_s	The summation intensity.
L	The Lorentz correction.
L_a	The attenuation length.
λ	A wavelength.
$\boldsymbol{m}_1$	A vector in the rotation coordinate system.
$\boldsymbol{m}_2$	The rotation axis.
$\boldsymbol{m}_3$	A vector in the rotation coordinate system.
μ_a	The linear attenuation coefficient.
$\boldsymbol{n}_p$	The polarisation normal.
P	The polarisation correction.
ϕ	The rotation about the ϕ axis.
size	The pixel size.
Q	The detector quantum efficiency.
R	The rotation matrix $R(\boldsymbol{m}_2, \phi)$.
$\boldsymbol{r}$	A reciprocal lattice vector.
$\boldsymbol{r}_0$	The reciprocal lattice point, $\boldsymbol{r}_0 = \boldsymbol{UBh}$.
ρ	The distance of the reciprocal lattice point from the rotation axis.
$\boldsymbol{s}$	A diffracted beam vector.
$\boldsymbol{s}_0$	The incident beam vector.
$\boldsymbol{s}_1$	The central impact diffracted beam vector.
σ_B	The number of standard deviations above the expected value of the index of dispersion of the background in spot finding.
σ_D	The shape of the spot.

σ_M	The shape of the spot.
σ_S	The number of standard deviations above the expected value of the pixels of the background in spot finding.
t_0	The detector thickness.
θ	The diffraction angle.
$\boldsymbol{U}$	The crystal orientation matrix.
$\sigma(I)$	The observed intensity.
$\sigma^2_{I_s}$	The variance on the summation intensity.
X_{mm}	The X position on the detector in mm.
X_{px}	The X position on the detector in pixels.
Y_{mm}	The Y position on the detector in mm.
Y_{px}	The Y position on the detector in pixels.

Background GLM specific symbols

$a(\boldsymbol{\beta})$	The Fisher consistency correction as defined in Equation 3.6.
$\boldsymbol{\beta}$	The vector of model parameters which are estimated from the quasi-likelihood algorithm.
c	The tuning constant specifying the robustness of the algorithm. Smaller values increase robustness of the algorithm.
μ_i	The estimated Poisson mean for the ith pixel, computed from the model as $\log(\mu_i) = \boldsymbol{X\beta}$.
ϕ	The dispersion. For a Poisson distribution, $\phi = 1$.
$\psi_c(r_i)$	The weights on the residuals as defined in Equation 3.5.
r_i	The residual for the ith pixel given by $r_i = \frac{y_i - \mu_i}{\sqrt{v_{\mu_i}}}$.
$\boldsymbol{U}$	The scoring function for the quasi-likelihood estimator.
v_{μ_i}	The variance for the ith pixel. For a Poisson distribution this is equal to the mean, $v_{\mu_i} = \mu_i$.
$w(\boldsymbol{x_i})$	The weights on each row of the design matrix. In our implementation these weights are equal to 1.
$\boldsymbol{X}$	The design matrix describing the generalised linear model. A row in the design matrix is given as $\boldsymbol{x_i}$; each row gives the explanatory variables for pixel, i.
y_i	The value of the ith pixel.
$\mathcal{I}$	The Fisher information matrix.

Ice ring background specific symbols

B	The estimated background.
$\boldsymbol{\beta}$	The vector of model parameters which are estimated from the quasi-likelihood algorithm. The jth parameter is given by β_j.
b_i	The background shape in pixel, i.
c_i	The number of counts in pixel i.

x_i	The element of X.
y_i	The transformed pixel value.
$\boldsymbol{\mu_i}$	The estimated value for the ith pixel. This depends on the value of the parameters as $\boldsymbol{\mu} = \boldsymbol{X\beta}$..
$\rho(r_i)$	The robust function of residuals.
r_i	The residual for the ith pixel given by $r_i = \frac{y_i - \boldsymbol{\mu_i}}{\sqrt{v_{\boldsymbol{\mu_i}}}}$.
v_i	The variance in the counts in pixel, i.
$\boldsymbol{W}$	The diagonal matrix of weights where the ith diagonal element is given by $w(r_i)$, the weight for the ith residual.
w_i	The weight of pixel, i.
$\boldsymbol{X}$	The design matrix describing the linear model. A row in the design matrix is given as $\boldsymbol{x_i}$; each row gives the explanatory variables for pixel, i.
$\boldsymbol{y}$	The vector of pixel values transformed using the Anscombe transform. The value of the ith pixel is given by y_i.

Stills profile modelling specific symbols

c_{tot}	The total pixel counts in a spot.
ϵ	The distance from the mean Ewald sphere, $\epsilon = \lvert\boldsymbol{s}_2\rvert - \lvert\boldsymbol{s}_0\rvert$.
κ	The scale factor for the Ewald sphere RLP product distribution.
$\boldsymbol{L}$	The lower triangular matrix of profile parameters.
λ_0	The mean wavelength.
$\boldsymbol{M}$	The covariance matrix of the reciprocal lattice point distribution.
$\bar{\boldsymbol{\mu}}$	The mean of the conditional distribution.
$\boldsymbol{\mu}$	The mean of the RLP distribution in the reflection specific coordinate system.
$\tilde{\mu}$	The mean of the marginal distribution.
$\boldsymbol{\mu}_{XY}$	The mean of the model in the XY axis.
$\boldsymbol{\mu}_Z$	The mean of the model in the Z axis.
$\boldsymbol{P}$	The covariance matrix of the product of the Ewald sphere and RLP distributions.
$\boldsymbol{p}$	The mean of the product of the Ewald sphere and RLP distributions.
$\boldsymbol{Q}$	The covariance matrix of the distribution of diffracted beam vectors.
$\boldsymbol{q}$	The mean of the distribution of diffracted beam vectors.
$\boldsymbol{r}_E$	A reciprocal lattice vector lying on the Ewald sphere.
$\boldsymbol{R}_e$	The rotation matrix into the reflection specific coordinate system.
$\boldsymbol{S}$	The observed 2D covariance matrix of the spot shape.
$\lvert\boldsymbol{s}_0\rvert$	The mean Ewald sphere radius $\mu_E = \lvert\boldsymbol{s}_0\rvert$.
$\boldsymbol{s}_2$	The laboratory space vector to the reciprocal lattice point.
$\bar{\boldsymbol{\Sigma}}$	The covariance of the conditional distribution.
σ_E^2	The variance of the Ewald sphere radius.

σ_λ^2	The wavelength variance.
$\mathbf{\Sigma}$	The covariance matrix of the RLP distribution in the reflection specific coordinate system.
$\tilde{\Sigma}$	The variance of the marginal distribution.
$\mathbf{\Sigma}_{XY}$	The covariance of the model in the XY axis.
$\mathbf{\Sigma}_Z$	The variance of the model along the Z axis.
$\bar{\boldsymbol{x}}$	The observed 2D position of the spot.

Chapter 1

Introduction

1.1 Macromolecular X-ray Crystallography

Structural biology is the study of the molecular structure and function of biological macromolecules. For decades, macromolecular X-ray crystallography (MX) has been the dominant method for the determination of such structures. The Protein Data Bank (PDB) (Berman, 2000) contains a database of biological structures determined by a variety of experimental methods. At the time of writing, 150593 structures have been deposited, of which, 134588 (89%) were determined by X-ray crystallography (PDB, 2019c). Despite the advancement of other experimental techniques in recent years, such as nuclear magnetic resonance spectroscopy (NMR), cryo electron microscopy (cryo-EM), micro electron diffraction (microED) and neutron diffraction (ND), X-ray crystallography remains the most widely used experimental technique for determining molecular structure (see Figure 1.1).

In macromolecular X-ray crystallography, molecules are purified and crystallised. These crystals are then exposed to an intense beam of X-rays. This produces a characteristic diffraction pattern of bright spots of varying intensity, which encodes information about the 3D structure of the crystallised molecule. The diffracted X-rays are recorded as a sequence of images by a 2D area detector, which then need to be analysed to determine the atomic structure of the macromolecule as represented by its electron density distribution.

In order to do this, the intensities of the diffraction spots are estimated from the diffraction images and processed to compute crystal structure factor amplitudes. Information about the structure factor phases is not recorded in the diffraction pattern on the detector; however, they can be determined indirectly by various methods *via* a process known as phase estimation. The electron density distribution within the crystal unit cell can then be computed from the crystal structure factors, and the atomic model placed to fit within the electron density to produce the 3D

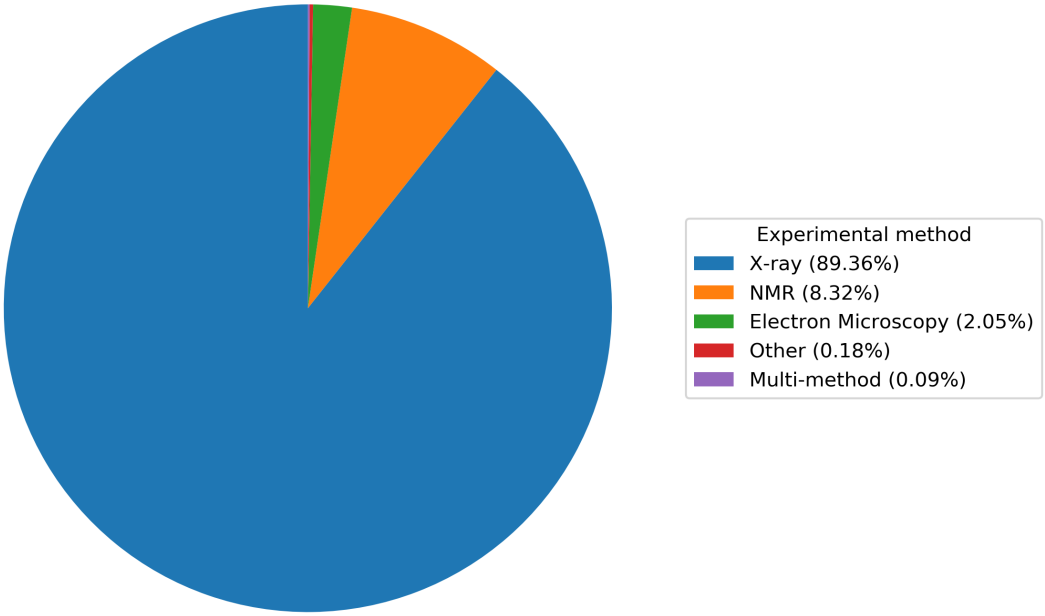


Figure 1.1: The percentage of structures deposited in the PDB solved using different experimental methods: X-ray diffraction was used in 89% of structure solutions (PDB, 2019b).

structure. This process is described in more detail in the following sections.

1.1.1 Data collection at synchrotron facilities

Synchrotrons have been used to perform MX experiments for over 40 years (Phillips *et al.*, 1976). In contrast to laboratory X-ray sources, synchrotrons have the advantage of being able to produce high brilliance monochromatic X-ray beams with tunable wavelengths (Helliwell and Mitchell, 2015). Since crystals of macromolecules tend to be small, and the strength of the diffraction is proportional to the illuminated volume of the crystal, high brightness X-ray beams are required to produce strong diffraction patterns to high resolution (James, 1948; Nave, 1989). Today, there are more than 100 MX beamlines at more than 20 synchrotrons around the world and more than 90% of new macromolecular structures determined by X-ray crystallography are solved using data collected at a synchrotron (Helliwell and Mitchell, 2015; Owen *et al.*, 2016; BioSync, 2019).

The most common experimental approach to the collection of X-ray diffraction data from a single crystal at modern synchrotron facilities is the rotation method (Arndt and Wonacott, 1977). In this method, the crystal is mounted on a goniometer between the X-ray beam and an area detector. During exposure, the crystal is rotated over a set angular range to collect a sequence of diffraction images until the desired volume of reciprocal space has been sampled.

For some macromolecules of interest, it is not possible to produce large enough and well diffracting crystals to perform a single crystal experiment. Since smaller crystals require a more intense X-ray beam in order to achieve the same strength of diffraction as a larger crystal, they are more susceptible to radiation damage. Radiation damage occurs when the sample absorbs photons from the incident beam resulting in the ionisation of atoms in the sample. It can induce both structural changes in the molecule and also causes changes to the crystal lattice (Garman, 2010). This limits the lifetime of the crystal in the beam and means that it may only be possible to collect a small wedge rotation. Consequently, data from multiple small wedges from multiple crystals must be combined in order to produce a complete dataset. Averaging the intensity measurements can reduce noise but can also reduce signal and add systematic errors due to differences resulting from non-isomorphism.

In recent years, the development of X-ray free electron laser (XFEL) facilities has led to a particularly extreme form of multi-crystal data collection known as Serial Femtosecond Crystallography (SFX) (Chapman *et al.*, 2011). The X-rays at an XFEL beamline are delivered in intense pulses where the pulse length is less than 100 femtoseconds. This allows a diffraction pattern to be collected at room temperature before any radiation damage occurs in the crystal; however, it also results in the crystal being destroyed after a single exposure, often termed "diffraction before destruction" (Neutze *et al.*, 2000). As a result, no rotation is possible and therefore each "still" image represents a single slice through reciprocal space. This method of data collection is also gaining traction at synchrotrons (Stellato *et al.*, 2014; Gati *et al.*, 2014; Owen *et al.*, 2017; Weinert *et al.*, 2017) where it is known as serial synchrotron crystallography (SSX).

1.1.2 Data reduction

The electron density at any point in space, $\rho(x, y, z)$, depends on the values of the crystal structure factors, F_{hkl}, of all recorded reflections such that, for the P1 space group:

$$\rho(x, y, z) = \frac{1}{V} \sum_{h} \sum_{k} \sum_{l} F_{hkl} e^{-2\pi i(hx+ky+lz)}. \tag{1.1}$$

Where, V is the crystal unit cell volume, (h, k, l) are the Miller indices and (x, y, z) are the points in real space. For higher symmetry space groups, the equation contains additional terms. The structure factors are complex valued; the amplitudes of the structure factors are related to the intensities of the Bragg spots, I_{hkl}, by:

$$|F_{hkl}| = \sqrt{K \frac{I_{hkl}}{LP}}. \tag{1.2}$$

Where, LP is the Lorentz-Polarisation correction and K is a constant. In order to infer the 3D positions of the atoms within the molecular structure from a set of diffraction images, the reflection intensities and associated errors first need to be estimated from the image pixel data. To do this, the unit cell and space group of the crystal need to be determined along with the experimental geometry in order to predict where on the diffraction images the Bragg spots are located. The intensity of the Bragg spots can then be estimated simply by summing the background subtracted pixel values corresponding to each reflection. The initial step of computing the raw uncorrected intensities from the diffraction images is known as "integration" and is performed by programs such as *MOSFLM* (Leslie, 1999), *XDS* (Kabsch, 2010b), *d*TREK* (Pflugrath, 1997), *HKL2000/DENZO* (Otwinowski and Minor, 1997) and *DIALS* (Winter *et al.*, 2018).

The raw reflection intensities are then processed to put them on a common scale and multiple measurements of the same reflection are merged; the intensities are then converted into structure factor amplitudes. This process is performed by a "scaling" program such as *POINTLESS/AIMLESS/CTRUNCATE* (Evans *et al.*, 2011; Winn *et al.*, 2011), *XSCALE* (Kabsch, 2010b) or *SCALEPACK* (Otwinowski *et al.*, 2003). The process of extracting reflection intensities, and consequently structure factor amplitudes, from the diffraction images is known as "data reduction"; A more thorough description of the data reduction is given in Chapter 2.

1.1.3 Phasing

The diffraction pattern on the detector records information about the structure factor amplitudes; however, the phase information is lost. In order to compute an electron density map, both the structure factor amplitudes and phases are required; therefore, the lost phases need to be determined indirectly. This is known as the *phase problem* in crystallography. For small molecules (with up to around 1000 atoms in the asymmetric unit), direct methods for determining the phases have been developed (Hauptman, 1997); however, these methods are generally not applicable to larger macromolecules, particularly as they require atomic resolution in order to find a phasing solution. An empirical rule-of-thumb is that if the resolution of the data is less than 1.2Å then direct methods are unlikely to yield a solution (Sheldrick, 1990; Morris and Bricogne, 2003): the average resolution of a structure deposited in the PDB is approximately 2.2Å (PDB, 2019a). The structure factor phases are then typically determined using one of the following methods.

1. *Molecular replacement (MR).* The method of molecular replacement (Rossmann, 1972) seeks to exploit non-crystallographic similarity between molecular structures to determine the phases. A homologous structure is used as a search

model and provides the initial phases for the unknown molecular structure. A rotation and translation search is then performed to find the best agreement between the observed diffraction from the unknown molecule and the predicted diffraction from the search model (Evans and McCoy, 2007). Traditional software programs such as *MOLREP* (Vagin and Teplyakov, 1997) use a least squares procedure to minimise the difference between the observed and predicted Patterson function (Patterson, 1934). More modern software programs such as *Phaser* (McCoy *et al.*, 2007) employ a Maximum Likelihood based approach which selects the model with the highest probability given the data. The majority, 77%, of new structures deposited in the PDB have been solved using molecular replacement (Wojdyr, 2019).

2. *Experimental phasing (EP).* If there are no homologous structures for the molecule of interest, or there is a homologue but a solution is not possible, experimental phasing methods must be used to determine the phases of the unknown structure. Experimental phasing methods attempt to determine the phases by introducing a perturbation to the structure factors *via* the introduction of a heavy atom, a wavelength change around an absorption edge, or from Bijvoet differences if anomalous scattering is significant. Various techniques for experimental phasing have been used such as; *single isomorphous replacement* (SIR) where a dataset is collected from a native protein and a single derivative which contains additional heavy atoms in the protein structure; *multiple isomorphous replacement* (MIR) which is similar to SIR but uses multiple derivatives; *single isomorphous replacement with anomalous scattering* (SIRAS) and *multiple isomorphous replacement with anomalous scattering* (MIRAS) which are similar to SIR and MIR respectively but also make use of information from anomalous differences. Today, the most commonly used experimental phasing method is *single-wavelength anomalous dispersion (SAD).* This method has the advantage that it only requires a single dataset collected at a wavelength near the absorption edge of a heavy atom present at a number of sites within the macromolecular substructure. The heavy atoms will scatter out of phase with the rest of the atoms in the molecule and analysis of the anomalous differences between pairs of reflections related by inversion enables the location of the heavy atoms within the substructure to be determined. This partial solution can then be used to determine the phases for the whole structure (Terwilliger *et al.*, 2016). A related experiment is known as *multi-wavelength anomalous dispersion (MAD)* which requires a number of datasets to be collected from the same crystal at different wavelengths. Many software packages are able to perform experimental phasing such as *Phaser* (McCoy

et al., 2007), *SHELX* (Sheldrick, 2007) and *SHARP* (Bricogne *et al.*, 2003).

1.1.4 Refinement and model building

Following the determination of the structure factor amplitudes and phases, an initial electron density map can be calculated. The atomic model can now be placed within the electron density in order to determine the 3D structure. A refinement program is used to fit a chemically sensible atomic model into the observed electron density whilst at the same time computing the best possible electron density map. However, in crystallography, the optimal fit of the atoms is complicated by the fact that there is generally a very small observation to parameter ratio. Therefore, in order to properly fit the atomic model to the data, refinement programs must be able to utilise as much information about the model and data as possible. Collecting high resolution data is, therefore, important; adding more observations increases the observation to parameter ratio. Prematurely cutting the resolution, and discarding possibly useful information, in order to "improve" data processing statistics is therefore unhelpful for the refinement.

Modern refinement programs such as *REFMAC5* (Murshudov *et al.*, 2011), *SHELXL* (Sheldrick, 2015), *BUSTER* (Blanc *et al.*, 2004) and *phenix.refine* (Afonine *et al.*, 2012) handle the observation to parameter ratio problem by utilising sophisticated maximum likelihood methods and incorporating prior information such as the protein sequence and physical knowledge of chemical bond lengths and bond angles within a Bayesian framework. As well as being able to incorporate prior information into the refinement, the use of Bayesian methods gives the additional advantage of allowing direct incorporation of a variety of sources of information, such as experimental phasing information, within a rigorous statistical framework. Agreement between the refined atomic model and the various sources of observed data is maximised through the optimisation of the atomic coordinates, atomic displacement parameters and scale factors whilst ensuring these restraints are not violated (Murshudov *et al.*, 2011). To facilitate this, refinement programs often come with databases of known ligands and restraints. For the *REFMAC5* program, these restraints are generated by *AceDRG* (Long *et al.*, 2017).

As well as typically being the final step in the structure solution pipeline, refinement programs are also used as an intermediate step in order to improve partial structural models and improve the electron density for automated model building pipelines such as *ARP/WARP* (Langer *et al.*, 2008), *Buccaneer* (Cowtan, 2006) and *SOLVE/RESOLVE* (Terwilliger, 2003) which attempt to automatically trace the backbone of the molecule and place the atoms within the electron density. Refinement software is also incorporated into molecular graphics programs such as *COOT*

(Emsley *et al.*, 2010) in order to guide manual structural model updates. Given the observed and calculated structure factor amplitudes, $|F_{obs}|$ and $|F_{calc}|$, the quality of the refined structure is typically assessed by the crystallographic R-factor, given by:

$$R = \frac{\sum_h |F_{h,obs}| - |F_{h,calc}|}{\sum_h |F_{h,obs}|}. \tag{1.3}$$

This gives a measure of agreement between the refined model and the observed data. However, assessment based purely on this indicator can result in over-fitting of the data. Therefore, the free R factor (Brünger, 1992) is often used as well. A "free" set of reflections is selected after data reduction and not used in the subsequent phasing and refinement steps. These reflections are then used to compute the free R factor. Successful refinement will result in a reduction in the free R factor.

1.2 Integration software

Over the years, numerous integration programs have been written; in the field of X-ray crystallography, the fact that software has generally been well described in the literature has enabled the quick development of new programs implementing new features whilst taking advantage of previous developments. In the following sections, a brief overview of the software currently being used for both rotation experiments and serial crystallography experiments is given.

1.2.1 Programs designed for rotation data

MOSFLM

The *MOSFLM* (Leslie, 1999) program was the main integration program distributed with the CCP4 (Winn *et al.*, 2011) software suite. It has been widely used for the processing of data collected using the rotation method for more than three decades. In recent years, the indexing component of the program has also been used within serial crystallography pipelines (White *et al.*, 2016). It uses a 1D FFT algorithm for auto-indexing and a 2D profile fitting algorithm for integration. The program is written in FORTRAN with a graphical user interface, *iMOSFLM* (Powell *et al.*, 2017), written in Tcl/Tk. The software is open source.

XDS

The X-ray Detector Software (*XDS*) program (Kabsch, 2010b) has been developed for more than 30 years (Kabsch, 1988). It is best known for its use of the 3D profile fitting method utilising a reflection specific coordinate system to perform the profile

fitting in reciprocal space. It is written in FORTRAN and has been designed for high performance and parallelism and it is included in various automated data processing pipelines at synchrotrons (Winter, 2009). It also has a basic graphical user interface, *XDSGUI* (Diederichs, 2019). The software is free for academic users.

HKL2000/DENZO

The *HKL2000/DENZO* suite of data processing programs (Otwinowski and Minor, 1997) has been in development since the 1990s. It is notable for providing the first implementation of a FFT based auto-indexing algorithm. As with *MOSFLM*, it also implements a 2D profile fitting algorithm. The software requires a paid user license for use; it is especially popular at facilities in the USA, China and Japan.

EVAL15

The *EVAL15* integration program (Schreurs *et al.*, 2009) takes a unique approach to integration. The majority of data processing programs perform profile fitting by learning a set of reference profiles from the observed strong spots by empirically averaging the observed refection profiles. *EVAL15* performs an *ab initio* prediction of the 3D reflection profile of each reflection by drawing from a set of probability distributions that describe the experiment and constructing each reflection profile using a ray-tracing approach. The program is generally used to integrate datasets with pathologies that may cause problems for more traditional data processing programs (Porta *et al.*, 2011). The software is free for academic users.

1.2.2 Programs designed for still data

cctbx.xfel

The Computational Crystallography toolbox (*cctbx*) is a collection of libraries for performing many routine tasks in crystallography. *cctbx.xfel* (Brewster *et al.*, 2016) is the data processing module within the *cctbx* project for the data processing and reduction of serial crystallographic data. The software was originally used to analyse XFEL data but can also be applied to serial crystallographic data collected on a synchrotron. The software is written in Python and C++ and is open source.

CrystFEL

CrystFEL is a pipeline for the processing of still X-ray diffraction data (White *et al.*, 2016). It was designed for processing data from XFELs but can also be used for synchrotron serial diffraction data. The program is possibly the most widely used

data processing program for the processing of still X-ray diffraction data. The software is open source.

Cppxfel

Cppxfel is a set of programs for processing still X-ray diffraction data (Ginn *et al.*, 2016). It was designed to be a test-bed for new algorithms being developed for data collected at XFELs. The algorithms developed in *Cppxfel* have been implemented in other software packages such as *DIALS* and *CrystFEL*. The software is open source.

nXDS

The *XDS* program was adapted for use with serial crystallographic data; the resulting program is called *nXDS* (Kabsch, 2014). The program adapts the 3D profile fitting algorithm used in *XDS* for use with still images. The software is free for academic users.

1.2.3 The *DIALS* framework

The Diffraction Integration for Advanced Light Sources (*DIALS*) project (Winter *et al.*, 2018) aims to develop an extensible, modular framework for the development of integration programs for macromolecular X-ray crystallography. A key objective is to produce a software package that is capable of performing the data reduction and analysis of X-ray diffraction data from both rotation and stills experiments in a consistent way within the same framework. The software is written in a combination of Python and C++. The high level interface source code is written in Python and time critical algorithms requiring high-performance are written in C++. Since the software is written in a modular way, new algorithms can be readily added with minimal effort. The algorithms developed in this project will, therefore, be implemented within the *DIALS* framework. *DIALS* is incorporated into the *xia2* (Winter, 2009) automatic data processing pipeline and is, therefore, effectively run on every MX dataset collected at Diamond Light Source.

1.3 Project motivation

1.3.1 New challenges in Crystallography

Various challenges facing structural biologists are driving the development of new technology and new modes of data collection in macromolecular crystallography. The development of advanced light sources such as XFELs has popularised "still" image data collection strategies, both at XFELs and at synchrotrons. At the same time,

the widespread automation of data acquisition on synchrotron beamlines and the development of pixel array detectors, with their associated fast readout times and near zero readout noise, have changed the way in which users interact with their data. There is an urgent need for fast data processing software that is able to keep up with the ever increasing speed of data collection, and that is able to provide accurate intensity estimates from weak diffraction data.

1.3.2 Micro crystallography

During purification and crystallisation of the macromolecule, it may be difficult or impossible to grow a crystal of sufficient size to perform a single crystal X-ray diffraction experiment; however, it may be possible to grow many tiny micro crystals from which it may be possible to collect a limited amount of data. Since the reflection intensities are proportional to the size of the illuminated volume of the crystal, diffraction data from micro-crystals is often very weak and noisy. In order to achieve the same diffraction strength as for a larger crystal, the incident beam intensity must be much higher. However, this also makes micro crystals much more susceptible to radiation damage during data collection. As a result, it is necessary to design new detector technology and data collection methods to collect data of sufficient quality to solve the molecular structure from multiple weakly diffracting crystals.

1.3.3 Pixel array detectors

Throughout the history of X-ray crystallography, changes in detector technology have precipitated changes in data collection methodology and data processing software. A prominent recent development in X-ray detector technology has been the development of pixel array detectors, such as the DECTRIS PILATUS detector (Henrich *et al.*, 2009). Compared with previous generations of detectors, such as charge-coupled devices (CCDs) and image plate detectors, photon counting pixel array detectors offer a drastic reduction in the read out noise that can be achieved. Pixel array detectors are composed of a 2D array of photon counting chips that operate independently of one another; each individual detector pixel is essentially a stand-alone X-ray detector; therefore, the point spread function is very small compared with CCD detectors. Photon counting pixel array detectors operate by counting individual X-ray photons as they impact on the detector. When a photon is absorbed by the sensor, it causes a charge pulse; if the pulse exceeds a certain threshold then the counter is increased. In this way, even single photons can be detected. Since the electronic noise in the sensor is low compared with the threshold, the images are essentially noise free. Another advantage that these detectors offer is fast readout times with maximum

frame rates for the next generation DECTRIS EIGER detector being 3000Hz with a 3.8μs dead time (Casanas *et al.*, 2016). The combination of fast readout times and low readout noise has changed the way that data is collected in MX; extremely fine-sliced datasets can be collected without any increase in noise and datasets can be collected in an incredibly short time period.

Whilst the quality of the data collected using these detectors is generally superior to data collected using older detectors, they do raise their own issues. The photon counts recorded in each pixel are Poisson distributed and do not need to be multiplied by a gain value to convert from analogue to digital units as in data collected with CCDs. The low read out noise of the detector also enables data to be collected with very low background. It is common to find implicit assumptions that the pixel counts are Normally distributed in algorithms within data processing programs; this is appropriate for data with a high background but wholly inappropriate when applied to low background data collected with a photon counting detector. The Poisson distribution may be approximated by the Normal distribution for a large number of counts but this approximation is inappropriate where counts are small. In this case, algorithms need to be modified in order to make correct statistical assumptions about the data. Photon counting detectors also have problems with a large photon flux; they have a count rate limitation which when exceeded can result in counts being lost. The detectors implement a count rate correction; however, this also has an error associated with it. Finally, pixel array detectors have gaps between the sensor chips which are handled through the use of "virtual pixels". Along a row of pixels, the gap between sensor chips has the same width as a single pixel and is spanned by large pixels either side of the gap with a collecting area 1.5 times the collecting area of a normal pixel. The counts from the two larger pixels are then distributed into three virtual pixels after readout. Each large pixel contributes two thirds of its counts to itself and one third of its counts to the gap pixel. This means that the counts in the three virtual pixels are correlated. This can make it appear that background pixel counts are under-dispersed relative to a Poisson distribution. All these issues need to be addressed by data processing programs.

1.3.4 Serial synchrotron crystallography

A prominent development in the field of X-ray diffraction has been the emergence of X-ray free electron lasers (XFEL); this in turn has popularised an experimental technique known as serial femtosecond crystallography (SFX) (Chapman *et al.*, 2011). In this technique, diffraction from a single crystal results in a single "still" diffraction image representing a single slice through reciprocal space; many thousands of diffraction images are required to produce a complete dataset, each of which may contain

diffraction from one or more crystals. This mode of data collection has been adapted for use at synchrotrons; a development known as serial synchrotron crystallography (SSX) (Stellato *et al.*, 2014). In the context of synchrotron experiments, the term "serial crystallography" takes on a slightly broader meaning, encompassing both "still" diffraction images, as in XFEL experiments, and individual small rotation images. When performing a SSX experiment, it may be generally preferable to collect individual small rotations rather than still images since small rotations allow greater coverage of reciprocal space (Hasegawa *et al.*, 2017); however, there are cases when it is preferable to collect still images. For example, a fixed target setup, collecting still diffraction images, allows users to perform the same experiment on both a synchrotron beamline and an XFEL beamline. This then enables incremental dose experiments to be performed in a time efficient manner at the synchrotron and allows the direct comparison of dose-resolved SSX and radiation damage-free XFEL structures of radiation sensitive proteins (Ebrahim *et al.*, 2019).

Processing X-ray diffraction data from SFX and SSX experiments has a specific set of challenges associated with it; as a result, new integration software has been developed and established software, designed for rotation experiments, has had to be modified (Kabsch, 2014; Kroon-Batenburg *et al.*, 2015; Brewster *et al.*, 2016; Ginn *et al.*, 2016; White *et al.*, 2016). The difficulties in processing still image X-ray diffraction data are numerous and related to the fact that each image only represents a single thin slice through reciprocal space and each image contains diffraction from a different crystal (or crystals). One of the defining characteristics of this data is, therefore, that all the reflections are "partially" recorded and, consequently, all the intensity measurements represent some fraction of the true reflection intensities. This impacts everything from the indexing and refinement, to the integration and scaling. Therefore, handling still diffraction data requires modification and special attention in almost every aspect of a data processing program. New algorithms are needed to improve the data processing for SSX to make it as robust, reliable and user friendly as data processing for rotation data.

1.3.5 Project aims

The aim of this project is to develop statistically robust methods for the integration and analysis of X-ray diffraction data being produced by new technologies, such as pixel array detectors, and new methods of data collection, particularly SSX, which is increasingly being used at synchrotrons. New methods and new technologies bring with them new challenges for data processing software. The algorithms developed as part of the project to address these challenges are incorporated into the *DIALS* integration package; this allows the algorithms to be readily used to process data

collected on MX beamlines at Diamond Light Source and at other synchrotrons that include *DIALS* within their data processing pipelines. In the spirit of the *DIALS* paradigm, the algorithms are developed and implemented in a modular way, allowing many different algorithms to be tested and deployed as they become available.

The primary focus of this project is on data collected from synchrotron sources using pixel array detectors, such as the PILATUS detectors used at Diamond Light Source. This work is timely, since these types of detectors are now the norm and assumptions made about the statistical properties of the data collected using CCD detectors may no longer hold. In this project, algorithms are developed and implemented making appropriate assumptions about the statistics of the data. The development of better algorithms will facilitate information extraction from extremely weak and noisy data that is a feature of many challenging structural biology problems.

This project has been carried out jointly at the Laboratory of Molecular Biology (LMB) at the University of Cambridge and at Diamond Light Source (DLS), the UK's national synchrotron facility. A major advantage of conducting this research and software development at a synchrotron facility is that it gives access to a wealth of experimental data that can be used (with the appropriate permissions) for testing and to guide algorithmic development. With the move to remote data collection and ever greater automation on synchrotron beamlines, crystallographic data processing is increasingly being done at the beamline rather than in the lab by individual users. Therefore, developing the software at a synchrotron facility enables faster deployment of the software where it is most needed: at the beamline.

Chapter 2

Integration of X-ray diffraction data

2.1 Introduction

The essential structure of a diffraction data processing program for X-ray crystallography has not changed much in over 40 years. Nyborg and Wonacott (1977) described three systems in common use at the time: the Cambridge System (Nyborg *et al.*, 1975; Ford, 1982), the Harvard System, and the Munich System (Schwager *et al.*, 1975). These computer programs all performed the same basic procedure: taking approximate values for the unit cell and experimental geometry as input, performing a refinement of the parameters, predicting the location of the diffraction spots, and then finally integrating them. The Cambridge System achieved arguably the most enduring success, being the predecessor to the popular *MOSFLM* integration program (Leslie, 1999) which has been in widespread use from that time until the present day. A number of other integration programs have since been developed and used within the field, most notably *XDS* (Kabsch, 1988), *d*TREK* (Pflugrath, 1999) and *HKL2000/DENZO* (Otwinowski and Minor, 1997).

The work described in this thesis is implemented within the *DIALS* framework (Waterman *et al.*, 2013; Winter *et al.*, 2018). The *DIALS* project was initiated with the aim of writing new integration software to address new challenges within the field resulting from the development of pixel array detectors, the popularisation of new modes of data collection, such as serial femtosecond crystallography (SFX) and serial synchrotron crystallography (SSX), and to exploit new computing infrastructure to enable data processing to proceed on the same timescale as data collection. Following the methodology of the *cctbx* (Grosse-Kunstleve *et al.*, 2002), *DIALS* is a hybrid system written in C++ and Python. Python lends itself well to rapid development, with an emphasis on clean, portable code, and has an extensive standard library. Various language features facilitate the easy implementation of generic code with interchangeable components. There is, however, a performance overhead with the

use of Python, due to the interpreted nature of the language, so performance critical code is implemented in C++. The *boost.python* (Abrahams and Grosse-Kunstleve, 2003) language binding framework is used to export the C++ interface for use in Python.

There is an explicit separation of the various data processing steps within the *DIALS* framework. In order to ensure modularity, each step is contained within its own command line program. The data flow within the *DIALS* framework is shown in Figure 2.1 and each data processing step is described as follows:

1. *Interpretation of image metadata*: The first step is to construct an initial model of the experimental geometry. This can be done manually; however, it is far more convenient to initialise the experimental geometry from the metadata contained within the headers of the diffraction images.

2. *Spot finding*: The diffraction images are then read and processed to extract a list of coordinates and approximate intensities of strong spots observed on each image.

3. *Indexing*: Using the initial experimental geometry and list of strong spots, the basis vectors of the crystal lattice are determined along with the crystal orientation; Miller indices are then assigned to the strong spots.

4. *Refinement*: Using the indexed strong spots, the initial experimental geometry, and the crystal model determined during indexing, the experimental geometry is optimised to determine a model that better predicts the location of the Bragg spots on the images. This is done by minimising the sum of the squared residuals between the observed and predicted positions of the spots on the detector.

5. *Integration*: The refined model of the experimental geometry is then used to predict where the Bragg spots will be recorded on the sequence of recorded images. The pixels associated with the Bragg spots are then extracted from the images and their intensities are estimated along with an estimate of the error on the intensities.

6. *Scaling*: The raw reflection intensity estimates from the integration step are then analysed to put intensities of symmetry equivalent reflections on a common scale. The scaled intensities are then merged and averaged for each symmetry equivalent reflection.

This chapter will describe each step of the data processing pipeline and an implementation in the context of the *DIALS* project (Winter *et al.*, 2018).

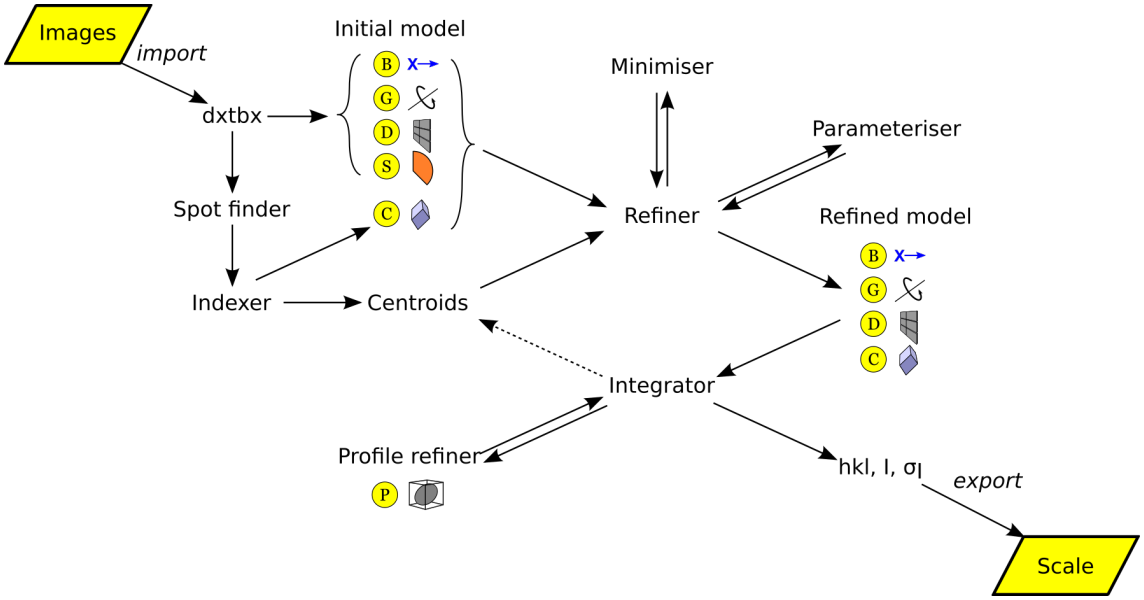


Figure 2.1: The data flow within the *DIALS* framework. The beam, crystal, detector, goniometer, profile and scan models are indicated by the yellow circles; hkl, I and σ_I are the miller indices, intensities and errors on the intensities respectively. The initial experimental geometry is read from the image metadata and strong spots are found on each image. A crystal model is then found through auto-indexing and Miller indices are assigned to each strong spot. The experimental geometry is then refined using the difference between observed and predicted strong spot positions. The positions of all Bragg spots are then predicted and their intensities are estimated. Finally the raw intensities of symmetry equivalent reflections are put on a common scale and averaged.

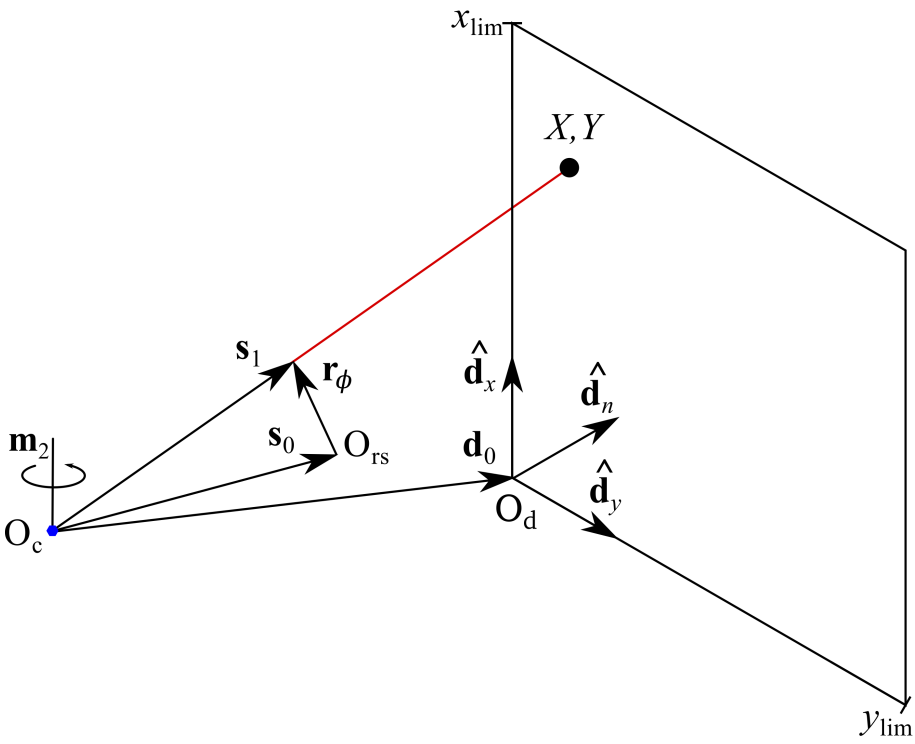


Figure 2.2: The description of diffraction geometry for the rotation method. A monochromatic X-ray beam is represented by the wave-vector $\boldsymbol{s}_0$, which intersects a sample rotation axis, given by the unit vector $\boldsymbol{m}_2$, at the origin of the laboratory coordinate system, O_c. An abstract detector plane is described in the real space laboratory coordinate system with an origin vector $\boldsymbol{d}_0$ and a pair of orthogonal basis vectors $\left\{\hat{\boldsymbol{d}}_x, \hat{\boldsymbol{d}}_y\right\}$. Here, O_d is the origin of the detector. The detector model provides a pair of limits, X_{lim} and Y_{lim}, forming a bounded rectangular panel within the plane. A crystal model has its setting expressed in a ϕ-axis frame (aligned to the reciprocal laboratory frame with origin, O_{rs}, at a rotation angle of $\phi = 0°$) by the setting matrix $\boldsymbol{UB}$, following the PDB convention. Diffraction is represented by the wave-vector $\boldsymbol{s}_1$, which may be extended to the point (x, y) at which it meets the detector panel, in the panel's coordinate frame.

2.2 Experimental geometry

The experimental geometry can be described using a fully vectorial description that expresses only the abstract geometry of the experiment and not other properties. No assumptions are made about the geometry besides the intersection of the beam with the crystal and rotation axis. In particular, the rotation axis is not assumed to be orthogonal to the direction of the beam in the representation of a rotation method scan. As the geometry consists of vector descriptions, in principle, their components may be expressed in any chosen coordinate system; however, within *DIALS*, the geometry is expressed using the standard imgCIF conventions (Bernstein and Hammersley, 2006). Figure 2.2 shows the abstract experimental geometry for the single crystal rotation experiment whose components are described in more detail below.

The geometry of a single detector panel k is conveniently expressed by the matrix,

$\boldsymbol{d}^k = \begin{pmatrix} \boldsymbol{d}_x^k & \boldsymbol{d}_y^k & \boldsymbol{d}_0^k \end{pmatrix}$. For panel k, the columns of the matrix are the panel basis vectors $\boldsymbol{d}_x^k$ and $\boldsymbol{d}_y^k$, augmented by the translation vector $\boldsymbol{d}_0^k$, locating the origin of the panel frame in laboratory space (Figure 2.2). The use of matrix $\boldsymbol{d}^k$ conveniently simplifies the equation for reflection prediction to a projection along a scattered direction to the detector plane, completely avoiding trigonometric functions in favour of matrix operations (Thomas, 1992).

The crystal geometry is defined by a right handed coordinate system with reciprocal lattice basis vectors $\boldsymbol{a}^*$, $\boldsymbol{b}^*$ and $\boldsymbol{c}^*$ such that a matrix, $\boldsymbol{B}$, the reciprocal space orthogonalisation matrix, is defined with columns, $\boldsymbol{B} = (\boldsymbol{a}^*, \boldsymbol{b}^*, \boldsymbol{c}^*)$. The orientation of the crystal is described *via* a rotation matrix, $\boldsymbol{U}$.

2.2.1 Prediction of Bragg spots

Given a Miller index, $\boldsymbol{h}$, crystal orientation matrix, $\boldsymbol{U}$ and the transpose of the crystal reciprocal space orthogonalisation matrix, $\boldsymbol{B}$, the vector to the reciprocal lattice point is given by:

$$\boldsymbol{r}_0 = \boldsymbol{UBh}. \tag{2.1}$$

Rotation of reciprocal lattice point onto Ewald sphere

The reciprocal lattice point $\boldsymbol{r}_0$ will result in an observed Bragg reflection if it passes through the Ewald sphere as shown in Figure 2.3. In a single crystal rotation experiment, this occurs at some rotation by an angle, ϕ, around the goniometer axis, $\boldsymbol{m}_2$ such that the rotated reciprocal lattice vector is given by $\boldsymbol{r}_\phi = R(\boldsymbol{m}_2, \phi)\boldsymbol{r}_0$. Indeed where such a rotation exists, the reciprocal lattice point will pass through the Ewald sphere twice: once as it enters the Ewald sphere and again as it exits. The angles at which the crystal needs to be rotated for the reciprocal lattice point to pass through the Ewald sphere can be calculated as follows (Kabsch, 2010a).

Given the goniometer rotation axis, $\boldsymbol{m}_2$, and the incident beam vector, $\boldsymbol{s}_0$, a right handed coordinate system for the goniometer, $(\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3)$ can be defined, as in Kabsch (2010a), such that

$$\begin{aligned} \boldsymbol{m}_1 &= \frac{\boldsymbol{m}_2 \times \boldsymbol{s}_0}{|\boldsymbol{m}_2 \times \boldsymbol{s}_0|} \\ \boldsymbol{m}_3 &= \boldsymbol{m}_1 \times \boldsymbol{m}_2. \end{aligned} \tag{2.2}$$

The distance of the reciprocal lattice point from the rotation axis is defined as $\rho = \sqrt{|\boldsymbol{r}_0|^2 - (\boldsymbol{r}_0 \cdot \boldsymbol{m}_2)^2}$. If $|\boldsymbol{r}_0| > 2|\boldsymbol{s}_0|$, then the reciprocal lattice point is at too high a resolution to be rotated onto the Ewald sphere with radius $|\boldsymbol{s}_0|$. Likewise,

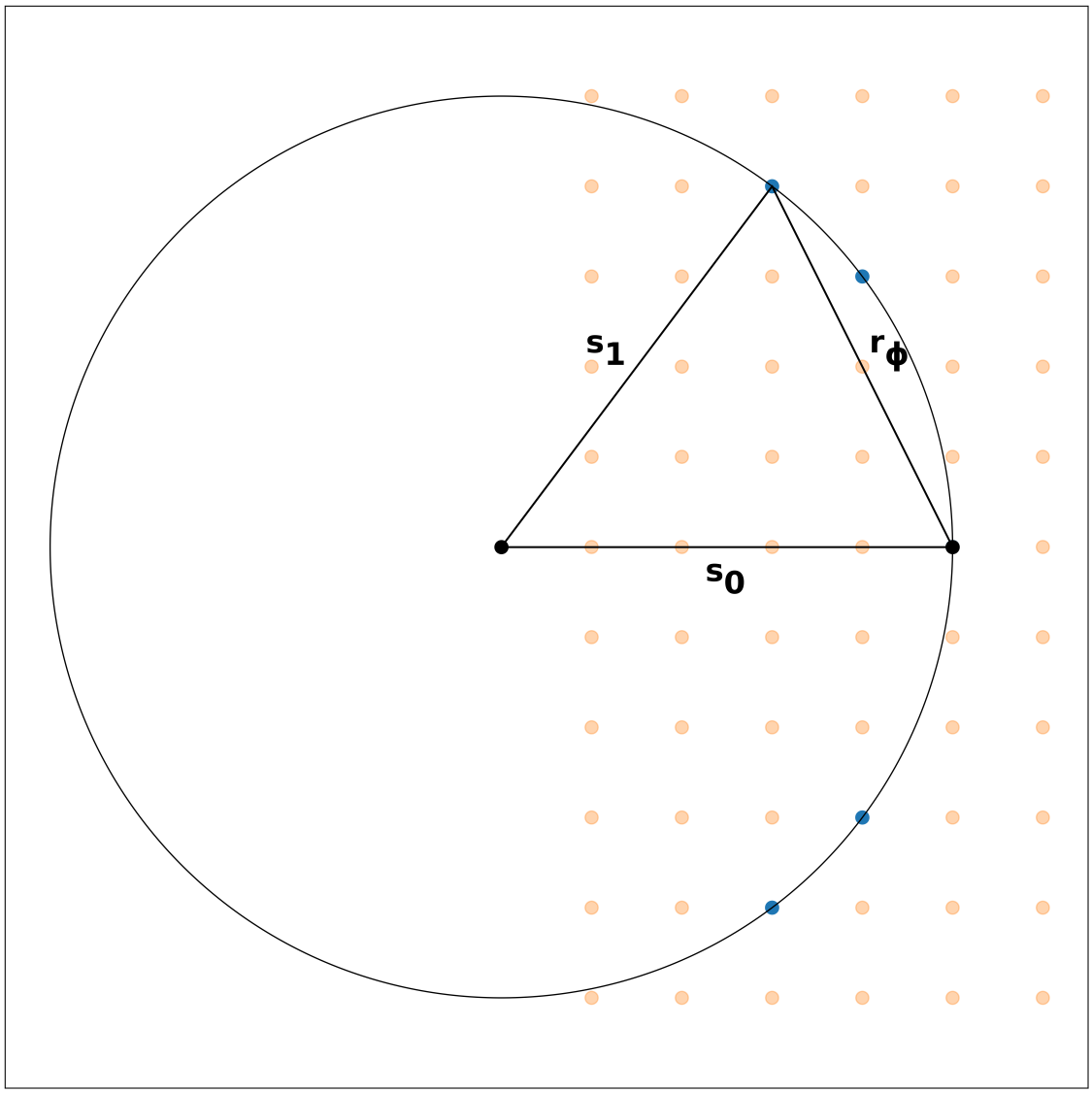


Figure 2.3: The Ewald sphere. The crystal is rotated in the beam with beam vector, $\boldsymbol{s}_0$, and a reciprocal lattice point, $\boldsymbol{r}_\phi$, enters the diffracting condition at the point that it passes through the Ewald sphere resulting in a diffracted beam vector, $\boldsymbol{s}_1$. In the figure, the reciprocal lattice points in and out of the diffracting condition are indicated by the dark and light circles respectively.

if $\rho^2 < (\boldsymbol{r} \cdot \boldsymbol{m}_3)^2$ then the reciprocal lattice point is in a blind region of reciprocal space and no rotation exists that will put the reciprocal lattice point onto the Ewald sphere. The components of the rotated reciprocal lattice vector in the goniometer coordinate system can be calculated as:

$$\begin{aligned} \boldsymbol{r} \cdot \boldsymbol{m}_3 &= -\left(\frac{\frac{|\boldsymbol{r}_0|^2}{2} + (\boldsymbol{r}_0 \cdot \boldsymbol{m}_2)(\boldsymbol{s}_0 \cdot \boldsymbol{m}_2)}{\boldsymbol{s}_0 \cdot \boldsymbol{m}_3} \right) \\ \boldsymbol{r} \cdot \boldsymbol{m}_2 &= \boldsymbol{r}_0 \cdot \boldsymbol{m}_2 \\ \boldsymbol{r} \cdot \boldsymbol{m}_1 &= \pm\sqrt{\rho^2 - (\boldsymbol{r} \cdot \boldsymbol{m}_3)^2}. \end{aligned} \tag{2.3}$$

The two angles at which the reciprocal lattice point passes through the Ewald sphere can then be calculated as follows:

$$\begin{aligned} \rho\cos(\phi) &= (\boldsymbol{r} \cdot \boldsymbol{m}_1)(\boldsymbol{r}_0 \cdot \boldsymbol{m}_1) + (\boldsymbol{r} \cdot \boldsymbol{m}_3)(\boldsymbol{r}_0 \cdot \boldsymbol{m}_3) \\ \rho\sin(\phi) &= (\boldsymbol{r} \cdot \boldsymbol{m}_1)(\boldsymbol{r}_0 \cdot \boldsymbol{m}_3) - (\boldsymbol{r} \cdot \boldsymbol{m}_3)(\boldsymbol{r}_0 \cdot \boldsymbol{m}_1) \\ \phi &= \tan^{-1}(\rho\sin(\phi)/\rho\cos(\phi)). \end{aligned} \tag{2.4}$$

Projection onto detector

The diffracted beam vector for a reciprocal lattice point rotated onto the Ewald sphere, $\boldsymbol{s}_1 = \boldsymbol{s}_0 + \boldsymbol{r}$ can be projected onto the detector as follows. Inverting the detector matrix, $\boldsymbol{d}$, results in a transformation matrix, $\boldsymbol{D} = \boldsymbol{d}^{-1}$; applying this transformation to the diffracted beam vector, $\boldsymbol{s}_1$, gives a point in projective space, $\boldsymbol{v} = \boldsymbol{D}\boldsymbol{s}_1$. For this to result in a valid projection of the reciprocal lattice point onto the detector, the vector component v_3 must be greater than 0. If the value of v_3 is negative, then the projection is a negative distance along the diffracted beam vector and does not impinge on the detector. The intersection of the diffracted beam vector on the detector on the virtual detector surface is given by

$$X_{mm} = \frac{v_0}{v_2}, \qquad Y_{mm} = \frac{v_1}{v_2}. \tag{2.5}$$

The conversion of this virtual detector surface coordinate, which by convention is given in millimetres, into a pixel coordinate on the detector is dependent on the model of the detector. If the pixels have a fixed size and all the photon energy is deposited at the surface of the detector then the pixel coordinates can be calculated as $X_{px}, Y_{px} = (X_{mm}/\text{size}), (Y_{mm}/\text{size})$. However, many detectors require additional corrections to compute the pixel coordinate. If the sensor has a finite thickness and the diffracted X-ray beam is not orthogonal to the detector surface, the pixel coordinate will depend on the mean distance at which the diffracted X-rays deposit

their energy as discussed below.

2.2.2 Parallax correction

The physics of direct conversion pixel array detectors, particularly those with a silicon sensor, gives rise to a small distortion of the diffraction image: the diffraction spots are elongated due to the passage of the photons through the sensor. This gives rise to a predictable effect on the central impact (Duisenberg *et al.*, 2003) of the reflection, which may be corrected by the "pixel to mm" mapping.

The absorption of photons in a material is given by the Beer-Lambert law. Specifically, the fraction of photons transmitted a distance x into a material with known linear attenuation coefficient, μ_a is given by

$$\frac{I(x)}{I_0} = \exp(-\mu_a x). \tag{2.6}$$

From this, it can be shown that for a sample of thickness, t, the attenuation length, L_a, the distance into the sample at which the mean absorption occurs, can be calculated as:

$$L_a = \frac{1}{\mu_a} - \left(t + \frac{1}{\mu_a}\right) \exp\left(-\mu_a t\right). \tag{2.7}$$

For a diffracted beam vector, $\boldsymbol{s}_1$, striking a detector with normal vector, $\hat{n}$, and thickness, t_0, the effective distance is $t = t_0/(\boldsymbol{s}_1 \cdot \hat{n})$. Therefore,

$$L_a = \frac{1}{\mu_a} - \left(\frac{t_0}{\boldsymbol{s}_1 \cdot \hat{n}} + \frac{1}{\mu_a}\right) \exp\left(-\frac{\mu_a t_0}{\boldsymbol{s}_1 \cdot \hat{n}}\right). \tag{2.8}$$

The corrected position for a predicted ray impinging on the detector with fast axis, $\boldsymbol{d}_x$, and slow axis, $\boldsymbol{d}_y$, is then

$$\begin{aligned} X_{mm}' &= X_{mm} + L_a(\boldsymbol{s}_1 \cdot \boldsymbol{d}_x) \\ Y_{mm}' &= Y_{mm} + L_a(\boldsymbol{s}_1 \cdot \boldsymbol{d}_y). \end{aligned} \tag{2.9}$$

2.3 Interpretation of image metadata

Effective processing of X-ray diffraction data from single crystal diffraction experiments relies on an accurate model of the experimental geometry, which in turn depends on the ability to read, with no loss of information, the wide variety of data formats used for X-ray diffraction experiments. While many experiments for macromolecular crystallography employ a simple geometry (rotation axis perpendicular to the direct beam, coincident with one detector axis and in which the "beam centre"

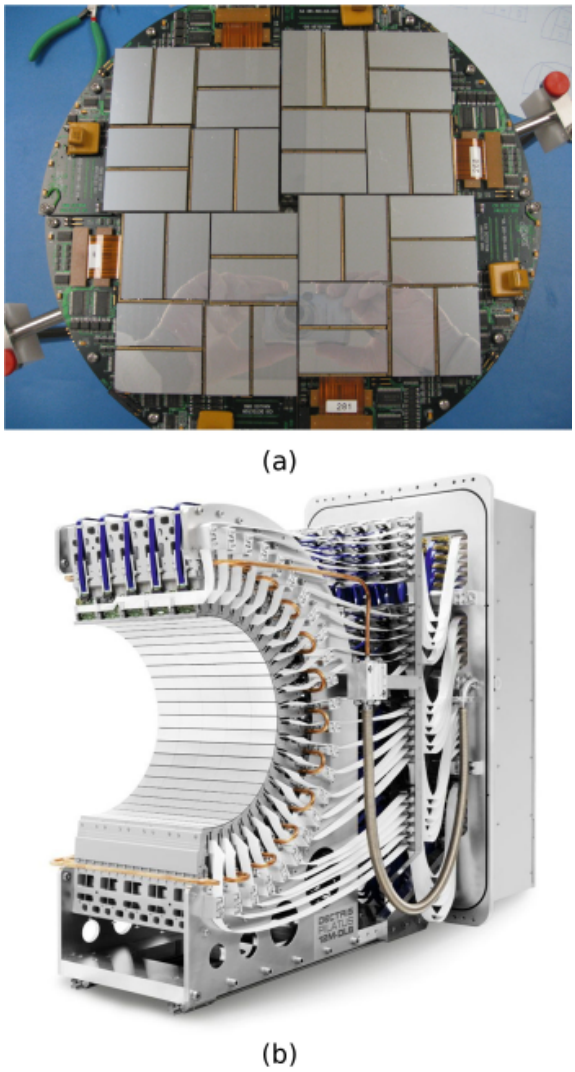


Figure 2.4: The CSPAD detector at the LCLS CXI beamline (a) and the PILATUS 12M-DLS at Diamond Beamline I23 (b) (Courtesy of DECTRIS Ltd).

is somewhere near the middle of the detector) the general diffraction experiment may employ a much more complex geometry, allowing for arbitrary positioning of a complex detector and the sample rotation axis. For example, the experiment may employ multi-axis goniometry or have a complex detector composed of multiple non-coplanar sensor panels (such as the PILATUS 12M-DLS used on Diamond beamline I23, Figure 2.4) (Wagner *et al.*, 2016). Reliable reproduction of this geometry from a range of different descriptions requires both a standardised representation and the ability to import the experimental geometry from a variety of instruments. This is complicated by the possibility of storing the information in different ways, *e.g.* expressing the beam centre in pixels or mm, or with different coordinate system conventions. While universal adoption of standards such as imgCIF (Bernstein and Hammersley, 2006) for the recording of X-ray diffraction data could resolve these challenges, historical precedent indicates that this is unlikely.

The task of developing a tool to uniformly read diffraction image headers and data has been addressed more than once. The CCP4 DiffractionImage library (Remacle and Winter, 2007) was developed to support the *DNA* (Leslie *et al.*, 2002) and *xia2* (Winter, 2009) projects, as it was realised early on that reliable access to a range of image headers was vital. This was, however, limited by a lack of extensibility and by assumptions made early in the design that the experimental geometry would correspond to the simple layout described above. The Computational Crystallography Toolbox (*cctbx*) (Grosse-Kunstleve *et al.*, 2002) includes a package, *iotbx.detectors*, providing data access for the indexing program *LABELIT* (Sauter *et al.*, 2004) and the XFEL data analysis program *cctbx.xfel* (Sauter *et al.*, 2013), yet it suffers similar limitations. More recent efforts, such as *FabIO* (Knudsen *et al.*, 2013) help to allow general access to the data but have less emphasis on the metadata so critical for crystallographic data and its analysis.

2.3.1 Implementation in *DIALS*

The diffraction experiment toolbox (*dxtbx*) is a software toolkit within the *cctbx* for writing new diffraction data visualisation and analysis applications, which has the aim of allowing a completely general and user-extensible approach to the reading and interpretation of diffraction image data and metadata (Parkhurst *et al.*, 2014). The *dxtbx* follows the principle that the interpretation and analysis of X-ray diffraction data should be distinct and separable. This design allows the *dxtbx* to be generally applicable to the reading of X-ray diffraction data and metadata and will help to liberate developers of data processing software from the often tedious task of supporting multiple file formats and data representations within their applications.

The *dxtbx* offers a general, user-extensible interface for the reading of X-ray diffraction data and provides abstract models in C++ and Python to describe the derived experimental geometry. For example, within the *dxtbx*, the geometry of a detector is expressed as a collection of abstract planes, each of which has a per-pixel mapping from the position on the surface to the pixel coordinates in the image. This mapping may be used to correct for static effects such as module position or CCD taper corrections, or for dynamic effects such as parallax correction in direct conversion detectors (described in more detail in Section 2.2.2). The interface exposed to the rest of the *DIALS* software is consistent, regardless of the underlying detector implementation, and has been used to treat data from new and complex detectors such as the CSPAD used for XFEL data collection at the Linear Coherent Light Source (Herrmann *et al.*, 2014; Brewster *et al.*, 2016), the DECTRIS PILATUS 12M-DLS used for long wavelength data collection (Wagner *et al.*, 2016) at Diamond Light Source beamline I23, and the HDF5-format (The HDF Group, 1997) of the

DECTRIS EIGER datasets (Casanas *et al.*, 2016).

The principle behind the *dxtbx* is to separate the interpretation of X-ray diffraction data from its analysis. Details of the experimental setup are encapsulated and exposed using a common interface and reference frame for all data types, ensuring that the client analysis code need not be aware of any file format specifics. The models produced by *dxtbx* describe the key experimental components and may be used directly, with no further transformation. *dxtbx* is also extensible in that a new experimental setup may be supported by the addition of a single Python file, that describes the local environment: once this has taken place no changes should be needed within *dxtbx* or the analysis code for the data to be correctly interpreted. Together these allow the developers of analysis code to focus on improving algorithms rather than the support of numerous detector data formats. Finally, the use of a completely general vectorial description of the experimental geometry allows for the propagation of detailed calibration information into the analysis code, and may also encourage analysis software to support a similarly general approach to the processing of X-ray diffraction data.

Library design

Early in the development of the *dxtbx*, it was recognised that, in order to be generally applicable, a library for reading diffraction image headers and data must satisfy the following requirements.

1. It must have the ability to read image data and metadata from a wide variety of detectors employing different file formats and experimental conventions.
2. The image data and metadata must be accessible *via* a single unified interface.
3. The library must be user-extensible without requiring modification of the library source code.
4. Finally, the models used to represent the experiment must be able to accurately capture the detector physics (*e.g.* distortion corrections) while being sufficiently general to capture a wide variety of diffraction measurement setups.

To achieve these aims, the *dxtbx* implements an extensible plugin framework, where beamline scientists and developers can add their own modules to handle input from different file formats with different file representations. At the cost of writing a small amount of Python code, the user may extend the library to support any bespoke file format and transform the metadata therein to correspond to the standard representation that is used within the *dxtbx* experimental models, which has been adopted from the imgCIF standard. A simple high-level interface that enables access to data from an entire sequence of images is also provided.

Experimental models

The *dxtbx* uses the concept of experimental models to encapsulate certain aspects of the experimental description that are separable with respect to one another. The experimental models are encoded in four container classes: the beam, goniometer, detector and scan. These contain information about the source wavelength and direction, the axis about which the crystal is rotated (for rotation data), the instrument performing the measurements, and relationship between the image frames and any rotation respectively. In the context of single crystal X-ray diffraction, the models are completely general with respect to experimental technique and beamline hardware. This is achieved by employing a fully vectorial description that expresses only the abstract geometry of the experiment and not other properties as described in Section 2.2.

Many ideas from the proposals described in the EEC Cooperative Programming Workshop on Position-Sensitive Detector Software (Bricogne, 1987) were used in the development of *dxtbx*. In particular, the scheme for "virtualisation" discussed therein, which involves forming an abstract and general definition for every component of the diffraction experiment, was adopted. The *dxtbx* forms the basis of the "instrument definition language" outlined at that workshop, by which actual beamline hardware is mapped to its abstract model representation for any particular experiment.

Of the core experimental models, the detector model is necessarily the most complex and requires further explanation. The basic unit of our abstraction is a panel, which represents a rectangular detector plane[1], oriented in laboratory space. The simple case, where the detector is a container of one or more such panels, none of which need to be co-planar, can accurately capture the half-barrel shaped PILATUS 12M-DLS constructed for Diamond Beamline I23 (Figure 2.4). For more exotic detectors, the *dxtbx* supports a general hierarchical model allowing panels to be organised into logically related groups and subgroups. This is necessary for the CSPAD (Hart *et al.*, 2012), used on the LCLS CXI beamline (Figure 2.4), where individual panels may move with respect to each other.

In determining a position on the detector, *dxtbx* uses the concept of a virtual detector plane. A position on the virtual plane is given by the panel identifier and a coordinate in the two-dimensional Cartesian frame attached to that panel. This point corresponds to the position at which photons impinge on the surface of the detector, and is independent of the actual detector hardware in use. Behind the virtual plane interface, the hardware-specific mapping between panel position and pixel location is encapsulated within a millimetre-to-pixel function (and its inverse),

[1]Currently *dxtbx* supports only detectors made of a collection of flat rectangular sensors; support for truly curved instruments, such as a Weissenberg image plate detector, could however be added when the need arises.

which must be supplied by code specific to the actual detector hardware. This will, for example, take into account detectors with thick sensors, where the interaction point within the sensor may alter the pixel position of the measurement. In *dxtbx*, this is realised by pairing the abstract detector model with a strategy class (Gamma *et al.*, 1994), that allows the behaviour of the detector model to be modified without changing the model itself. This class is the natural place for all hardware-specific distortions from the simple mapping, including parallax and geometrical distortion effects, for example caused by an optical fibre taper.

In general, all algorithms that use the *dxtbx* models do so *via* the vectorial representations summarised in Figure 2.2. This ensures that the choice of coordinate frame is independent of the working of those algorithms, with the caveat that the origin of the laboratory frame is located at the intersection of the primary beam and the sample.

Image metadata interpretation

A plugin mechanism is provided to handle input from multiple file formats with alternative descriptions of the experimental geometry with users or beamline staff able to add their own plugins to handle bespoke image formats or specific local variants. The ability to extend the library is primarily useful where either an unusual piece of experimental hardware is present, or if the beamline has some idiosyncrasies, for example a left-handed rotation axis. The plugin based model for handling different data representations has two advantages: no external site-file is required for operation and it enables complex corrections (*e.g.* tile position corrections for a PILATUS detector) to be encoded in a self-contained way.

Image metadata storage

A module is provided to enable straightforward storage of modified image set metadata. An image set may then be created from the file representation, allowing the refined experimental geometry to be saved for later use. The data are saved using the JavaScript object notation (JSON) format (Crockford, 2006); this format was chosen as it is human-readable, an open standard and is natively supported in many programming languages. In particular, the Python standard library contains a module for reading and writing arbitrary Python structures to JSON format, making it convenient for use within the *dxtbx*.

2.4 Spot finding

A set of observed spots is required in order to determine the unit cell and orientation of the crystal. Therefore, the first task performed by a data processing program is to analyse the raw X-ray diffraction images to extract a set of strong spots with well determined centroids for use in indexing. Since the initial experimental geometry may be poorly specified, spot finding algorithms typically do not require any experimental information beyond the diffraction images themselves. Spot finding algorithms tend to operate according to the following general procedure:

1. A pixel thresholding algorithm is applied to select strong pixels in the X-ray diffraction image.
2. A list of spots is then extracted by determining connected regions in the thresholded image (in two dimensions for still shots or three dimensions for rotation data).
3. The size, centre of mass and total intensity of the observed spots are calculated.
4. The resulting spot list is then filtered based on user criteria, *e.g.* minimum and maximum number of pixels in a spot, and the peak to centroid distance of the spot.

The method used for identifying strong pixels in the X-ray diffraction images differs in the various integration programs; the method used in *XDS* (Kabsch, 2010b) and *DIALS* (Winter *et al.*, 2018) is described here. The local mean, μ, and variance, σ^2, are calculated for each pixel (over the region around the pixel defined by the kernel size) in each image. The local index of dispersion, D, is then calculated from these quantities as

$$D = \left(\frac{\sigma^2}{\mu}\right). \tag{2.10}$$

For a detector with insignificant point-spread and gain G, a value of $D \approx G$ is expected for the background, with G being unity for a photon counting detector. The appropriate gain for integrating detectors is normally set by the relevant *dxtbx* format class, but if required the value can be modified for spot finding. Strong pixels are then identified through three sequential thresholding operations. First, pixels with a value less than a global threshold value (by default set to zero) are discarded. Next, a gain-dependent threshold is applied using the index of dispersion map to identify regions of the image that contain strong pixels. This operation essentially tests for regions of the image whose pixels are not drawn from a single Poisson distribution, *i.e.* not a local flat-field. For Poisson-distributed data, the quantity

$D(N-1)$ is approximately chi-squared distributed with $N-1$ degrees of freedom, where N is the number of pixels in the region (Frome, 1982). Therefore the expected variance in $D(N-1)$ is $2(N-1)$. Pixels are marked as potentially strong if the index of dispersion in a local region around the pixel is greater than a certain number of standard deviations, given by the parameter σ_B (by default set to 6.0), above the expected value,

$$D > G\left(1 + \sigma_B\sqrt{\frac{2}{N-1}}\right). \tag{2.11}$$

Finally, pixels in these regions are selected as strong if their raw values, c_i, are greater than a certain number of standard deviations, assuming a Poisson distribution, given by the parameter, σ_S (by default set to 3.0), above the local mean,

$$c_i > \mu + \sigma_S\sqrt{G\mu}. \tag{2.12}$$

This method will find features on the image, *e.g.* Bragg reflections, ice rings and zingers.

Implementation in *DIALS*

In some integration packages, the initial spot finding is limited to a subset of the data for the initial characterisation, *i.e.* indexing from a small number of images. Within *DIALS*, however, the decision was made to globally model the experiment. This has a significant effect on spot finding: the recommended usage (though this is not mandatory) is to find spots throughout the data set and perform subsequent indexing and refinement using this list of spots or a random subset. The spot list is also used to designate which reflections will contribute to the construction of reference profiles during integration.

For photon counting detectors, the default settings for the global threshold (0) and gain (1) are usually appropriate. For other detectors where these defaults are not correct, appropriate values can be set in the *dxtbx* library as part of the detector model, or manually adjusted during spot finding. In *DIALS*, determining appropriate parameters is easily accomplished *via* an interactive image viewer. The image processing steps in the spot finding algorithm are shown in Figure 2.5.

2.5 Indexing

Given a set of observed strong spots from a set of diffraction images, an auto-indexing program is able to determine the crystal unit cell parameters and orientation, and also assign Miller indices to each of the observed spots. In the past, indexing

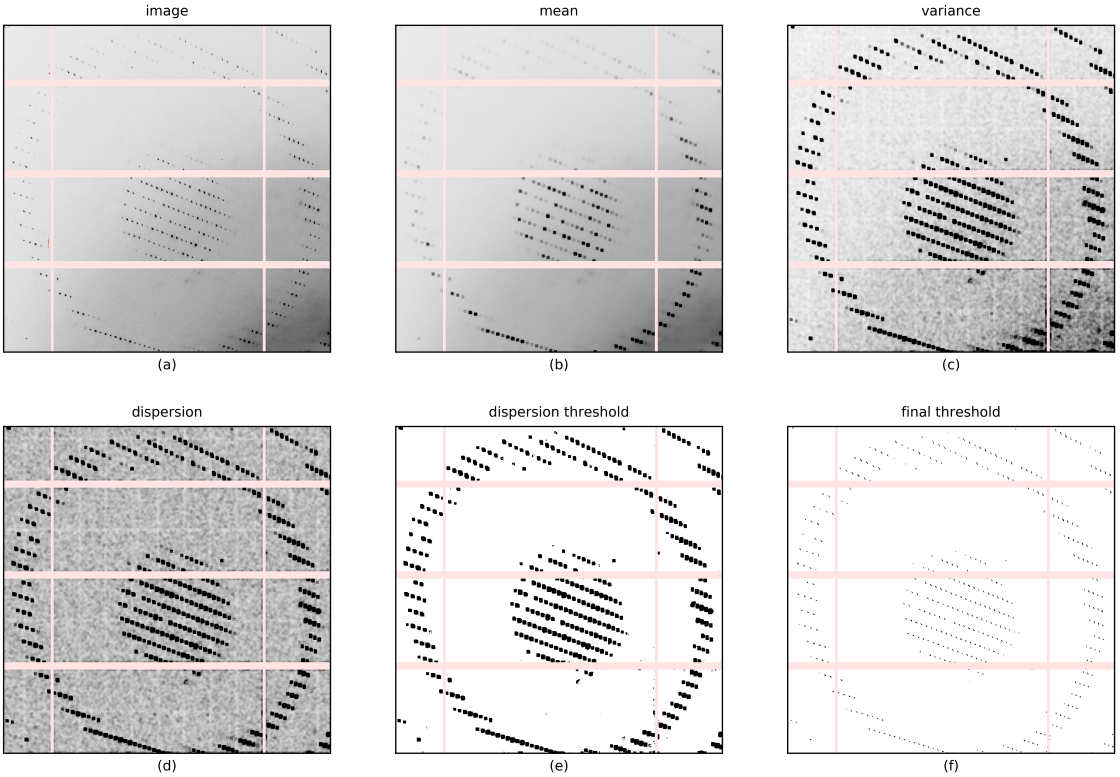


Figure 2.5: The image processing steps of the spot finding algorithm. The original image (a), the local mean (b) and variance (c) are calculated in a kernel centred on each pixel (by default, a square kernel of size 7×7 pixels is used). The local index of dispersion (d) is then calculated. A threshold (e) is applied to the index of dispersion image and this is combined with a threshold on the individual pixel counts (f).

algorithms typically needed prior information about the expected unit cell parameters (Messerschmidt and Pflugrath, 1987); however, most modern integration programs typically implement auto-indexing algorithms that are able to determine the unit cell parameters by direct analysis of the strong diffraction spots without any prior information. The only information required is a description of the experimental geometry: specifically, the wavelength of the incident beam is required along with a set of vectors describing the incident beam direction, detector position and detector orientation. The basic procedure performed by an auto-indexing program is as follows (Bricogne, 1986; Otwinowski and Minor, 1997; Powell, 1999; Sauter *et al.*, 2004; Gildea *et al.*, 2014):

1. For each spot in a set of strong spots, the 3D centroid is calculated, giving the position of the spot on the detector and the rotation angle at which it is observed. A diffracted beam vector for the central impact of the spot, and consequently a point in reciprocal space, can then be calculated.

2. The periodicity of the set of 3D points in reciprocal space is analysed and a set of basis vectors is selected to determine a likely primitive unit cell. A refinement of the experimental geometry can then be performed to minimise the distance between the observed and predicted positions of the spots; for rotation data, this takes into account the position of the spots on the detector as well as their rotation angle.

3. Transformations are then applied to select the compatible Bravais lattices and the experimental geometry is refined using the symmetry constraints for each Bravais lattice. Heuristics are then applied to select a Bravais lattice from the list; this typically involves selecting the highest symmetry solution which has a low penalty (Powell, 1999) and low RMSD between the observations and predictions.

4. Miller indices are then assigned to all the strong spots consistent with the lattice.

5. In the case of multiple lattices, the procedure can repeated iteratively with the remaining strong spots until no more spots can be indexed.

In order to determine the basis vectors of the reduced unit cell, auto-indexing programs search for periodicity in the reciprocal space mapping of the observed strong spots. It is in performing this task that the various implementations offered by auto-indexing programs tend to differ. Fourier transforms provide a natural framework for the analysis of periodic data; the use of a 3D Fourier transform of the reciprocal space points for determining the basis vectors of the reduced unit cell was

proposed by Bricogne (1986); however, performing a 3D fast Fourier transform (FFT) is very computationally intensive so application of this algorithm was not practical using the computer hardware available at the time. With advances in computing, a practical implementation of the 3D Fourier transform auto-indexing algorithm became feasible and the first implementation was provided by *HKL2000/DENZO* (Otwinowski and Minor, 1997; Otwinowski *et al.*, 2012). It was recognised that, it was much less computationally expensive to compute many 1D FFTs with a fine grid than a single 3D FFT with a much coarser grid. This precipitated the development of a 1D Fourier transform auto-indexing algorithm (Steller *et al.*, 1997; Rossmann and Beek, 1999; Powell, 1999). This has since become known as the DPS algorithm as it was first incorporated in the DPS software suite distributed with ADSC CCD detectors (Powell, 1999). The algorithm works by computing several thousand uniformly spaced projections of the spots in a hemisphere in reciprocal space. The basis vectors are extracted by finding the non-collinear directions showing the greatest periodicity.

Other, non Fourier transform based methods have also been developed. One algorithm was described by Kabsch (1993) for implementation in *XDS* (Kabsch, 2010a) which involved reducing the list of reciprocal space points into a set of difference vector clusters. Analysis of these clusters then enabled the extraction of the basis vectors. A particular challenge in recent years has been the processing of data containing multiple lattices on a single image. To address this, a real space grid search auto-indexing algorithm was developed in order to index spots derived from possibly multiple lattices (Gildea *et al.*, 2014). This algorithm requires prior knowledge of the unit cell parameters, meaning that only the orientation of the crystals needs to be determined from the data. Possible orientations are sampled uniformly within a hemisphere and scored to find combinations of basis vectors that are consistent with the input unit cell parameters.

Implementation in *DIALS*

The philosophy of the *DIALS* framework is that there is not necessarily a "best" algorithm for a particular task; consequently, the framework has been designed to allow various algorithm implementations to be used. Indexing provides a concrete example of this philosophy. *DIALS* offers three methods for determining the reciprocal lattice basis vectors which can be selected at runtime by the user (i) a 1D Fourier transform based algorithm, (ii) a 3D Fourier transform based algorithm, and (iii) a real space grid search algorithm (Gildea *et al.*, 2014).

Since auto-indexing requires knowledge of the experimental geometry, it is critical that this information is specified correctly. Initial geometry taken from the image metadata is often inaccurate; indeed, experience acquired from processing data from

a variety of sources has shown that the most common cause of indexing failure is the specification of an incorrect "beam centre" on the detector. An error in the beam centre greater than half the distance between adjacent reflections will inevitably result in mis-indexing of reflections. However, errors in the beam centre are generally problematic for indexing; the mapping of the spots to reciprocal space will become distorted, thereby making analysis of the periodicity difficult or impossible. In order to mitigate this problem, algorithmic solutions have been devised: Sauter *et al.* (2004) described a grid search algorithm that used information about the lattice spacings to derive a better estimate of the beam centre. This algorithm is also implemented within *DIALS* (Winter *et al.*, 2018).

2.6 Refinement

In the context of X-ray diffraction data processing, refinement is the process of optimising a set of experimental model parameters relating to the experimental geometry that affect the prediction of the positions of the reflections and their shape on the detector. There are three types of refinement implemented in data processing programs: centroid refinement, which aims to minimise the difference between the observed and predicted spot positions; profile refinement, which aims to model the shape of spots on the detector; and post refinement, which aims to enable more accurate estimation of the intensities of partially recorded reflections.

2.6.1 Centroid refinement

The initial experimental model supplied to the data processing program is derived from the image metadata, but this may not always be accurate. In order to estimate the reflection intensities, it is necessary to know their positions on the detector; therefore, to ensure accurate intensity estimates, the prediction of the spot positions must also be accurate. Accurate spot positions are calculated by performing a least squares optimisation that aims to find the set of parameters that results in the smallest sum of squared residuals between the observed and predicted spot positions (Kabsch, 2010a; Waterman *et al.*, 2016).

Implementation in *DIALS*

Over the course of a rotation scan, the experimental models may change. This is most evident in the crystal model: the unit cell volume may increase due to radiation damage or the orientation may change due to crystal slippage during rotation. These effects need to be incorporated into the experimental model for the prediction of reflection positions. Traditionally, refinement of the experimental model parameters

occurs during integration where small batches of images are processed together and a single set of parameters is refined for each batch of images (Kabsch, 2010a). However, this can result in discontinuities between refined parameters on adjacent images. Within *DIALS*, the refinement is performed as a discrete step before integration using information from the strong spots collected during spot finding across the whole scan range. The crystal model is parameterised such that smooth variation in the unit cell and orientation parameters is an explicit feature of the model (Waterman *et al.*, 2016).

DIALS also enables the joint refinement of multiple experiments. In this context, an "experiment" refers to a single set of experimental models that results in a diffraction pattern; for a rotation experiment, this means a beam, crystal, detector, goniometer and scan model; for a serial experiment, this means a beam, crystal and detector model. Experiments can share, for example, a detector or beam model as shown in Figure 2.6. In the case of multi-crystal experiments, where each experiment covers either a small rotation or consists of a single still diffraction image, the joint refinement can enable better determination of the detector position and reduce correlations between detector and crystal parameters. This then results in better determination of the crystal unit cell and orientation which consequently improves the predicted spot positions. Furthermore, in multi-crystal experiments, the quality of data is variable and merging a subset of datasets might give better results than merging all the acquired data. Since it is not generally computationally feasible to try all combinations of datasets, programs such as *BLEND* (Foadi *et al.*, 2013) select a subset by clustering the datasets in a hierarchical manner based on metrics derived from the unit cell parameters. Therefore, better determination of the unit cell parameters can result in a better clustering and consequently a better selection of datasets for merging.

2.6.2 Profile refinement

Once the positions of the reflections on the detector have been predicted, the shape of the reflections on the detector needs to be determined in order to enable the classification of pixels into foreground and background for each reflection. The shape of the spots on the detector is typically determined by specifying a model of the reflection profile in reciprocal space and estimating the parameters of the model from the observed strong diffraction spots on the images. The profile model used in *XDS* (Kabsch, 2010a) and *DIALS* (Winter *et al.*, 2018) uses a non-orthogonal coordinate system local to each reflection. For an incident beam vector, $\boldsymbol{s}_0$, and a given reflection with diffracted beam vector, $\boldsymbol{s}_1$, this coordinate system is defined as (Kabsch, 2010a):

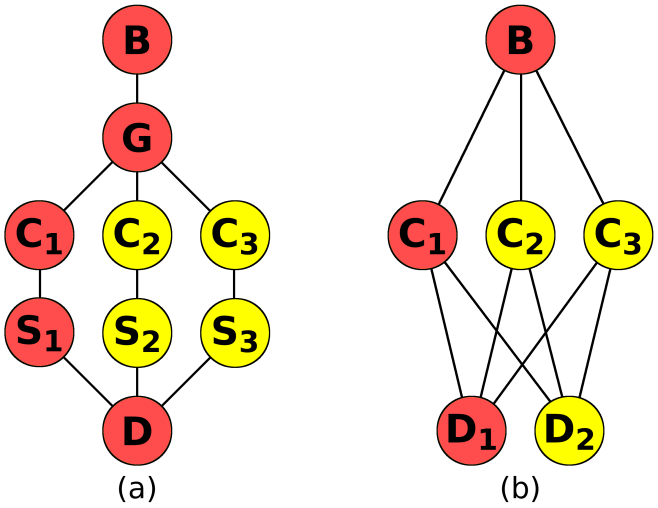


Figure 2.6: Two cases with multiple experiments. Here the circles indicate the beam (B), crystal (C), detector (D), goniometer (G) and scan (S) models. (a) A rotation experiment with three independent sweeps with different crystal and scan models but using the same beam, goniometer and detector model. (b) A serial crystallography experiment with a single beam and two detectors and three independent crystal models.

$$\begin{aligned} \boldsymbol{e}_1 &= \frac{\boldsymbol{s}_1 \times \boldsymbol{s}_0}{|\boldsymbol{s}_1 \times \boldsymbol{s}_0|}, \\ \boldsymbol{e}_2 &= \frac{\boldsymbol{s}_1 \times \boldsymbol{e}_1}{|\boldsymbol{s}_1 \times \boldsymbol{e}_1|}, \\ \boldsymbol{e}_3 &= \frac{\boldsymbol{s}_1 + \boldsymbol{s}_0}{|\boldsymbol{s}_1 + \boldsymbol{s}_0|}. \end{aligned} \tag{2.13}$$

The $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ axes form a tangent plane on the surface of the Ewald sphere centred on the intersection of the diffracted beam vector, $\boldsymbol{s}_1$, with the Ewald sphere. This component of the transformation corrects for geometrical distortion as a result of the obliquity of the incidence of the diffracted rays on the detector. The non-orthogonal $\boldsymbol{e}_3$ axis corrects for the path taken by the reflection as it is rotated through the Ewald sphere. The angle of rotation about the fixed rotation axis required for a reflection to pass through the Ewald sphere is a function of its resolution and position relative to the fixed rotation axis. High resolution reflections and reflections further away from the rotation axis will pass through the Ewald sphere faster than lower resolution reflections and reflections closer to the rotation axis. The shortest path through the Ewald sphere is taken when the rotation axis is orthogonal to both the incident and diffracted beam vectors. The increase in path length of the reflection through the Ewald sphere is $1/(e_1 \cdot m_2)$. Transformation into this coordinate system makes reflections appear as if they had taken the same path through the Ewald sphere.

For a pixel with observed diffracted beam vector, $\boldsymbol{s}'$, and rotation angle, ϕ', the transformation to a point, $(\epsilon_1, \epsilon_2, \epsilon_3)$, in this coordinate system is as described in Kabsch (2010a):

$$\begin{aligned} \epsilon_1 &= \frac{\boldsymbol{e}_1 \cdot (\boldsymbol{s}' - \boldsymbol{s}_1)}{|\boldsymbol{s}_1|}, \\ \epsilon_2 &= \frac{\boldsymbol{e}_2 \cdot (\boldsymbol{s}' - \boldsymbol{s}_1)}{|\boldsymbol{s}_1|}, \\ \epsilon_3 &= \zeta(\phi' - \phi). \end{aligned} \tag{2.14}$$

Where, $\zeta = \boldsymbol{m}_2 \cdot \boldsymbol{e}_1$. The reflection profile model is then a Normal distribution in this reflection specific coordinate system such that:

$$P = \frac{1}{\sqrt{2\pi}\sigma_D} \exp\left(-\frac{1}{2}\frac{\epsilon_1^2}{{\sigma_D}^2}\right) \frac{1}{\sqrt{2\pi}\sigma_D} \exp\left(-\frac{1}{2}\frac{\epsilon_2^2}{{\sigma_D}^2}\right) \frac{1}{\sqrt{2\pi}\sigma_M} \exp\left(-\frac{1}{2}\frac{\epsilon_3^2}{{\sigma_M}^2}\right). \tag{2.15}$$

Where the parameter, σ_D, therefore, determines the extent of the reflection on the face of the detector, and the parameter σ_M determines the extent of the reflection over a range of images. These parameters are estimated from the list of indexed strong spots identified previously during spot finding as described in Kabsch (2010a). Using this model, partiality of a reflection, the fraction of the total intensity of the reflection recorded on the detector images, is calculated as follows:

$$\text{partiality} = \frac{1}{2}\left[\text{erf}\left(\frac{|\zeta|(\phi_1 - \phi_c)}{\sqrt{2}\sigma_M}\right) - \text{erf}\left(\frac{|\zeta|(\phi_0 - \phi_c)}{\sqrt{2}\sigma_M}\right)\right]. \tag{2.16}$$

Where, ϕ_0 and ϕ_1 are the rotation angles covered by the image, and ϕ_c is the rotation angle for the predicted location of peak of the Bragg reflection.

Implementation in *DIALS*

The default profile model used in *DIALS* is the same as that used in *XDS*. The σ_D parameter is calculated as described in Kabsch (2010a) by computing the intensity weighted variance of the diffracted beam directions of the pixels contributing to the observed spot and then computing the mean of the variances over all strong spots. The σ_M parameter is computed *via* a maximum likelihood estimator which extends the method in Kabsch (2010a) and takes into account the number of images over which the strong spots are recorded as well as the fraction of the reflection intensity recorded on each image.

2.6.3 Post refinement

Post refinement is performed after integration in order to provide better estimates of the partiality for reflections which have not been fully recorded. As a crystal is rotated, reflections pass through the Ewald sphere. Due to the effects of mosaicity, the reflection covers a finite size in reciprocal space. A fully recorded reflection will pass fully through the Ewald sphere; a partially recorded reflection will only be rotated partially through the Ewald sphere. In the case of a partially recorded reflection, only a fraction of the total expected intensity will be recorded. This fraction can be calculated by applying a model of the reciprocal space reflection and computing the fraction of the total volume of the spot that has been rotated through the Ewald sphere. However, this can only be achieved if the unit cell parameters and crystal orientation are known to a high accuracy. Post-refinement uses the integrated and merged intensity to compute observed partialities; the unit cell parameters and crystal orientation are then refined to minimise the difference between the observed and expected partialities (Rossmann *et al.*, 1979; Winkler *et al.*, 1979). In *DIALS*, post refinement is not performed for rotation data.

In the processing of "still" X-ray diffraction images, the handling of partial reflections is of additional importance. In this method, all reflections are partially recorded and the recorded intensity essentially represents a slice through the reciprocal space reflection profile. Calculating the fraction of observed intensity in this case is non-trivial since the volume of the reflection profile that is rotated through is always zero. Various models for computing the partiality of still diffraction images and performing post-refinement have been proposed (Hattne *et al.*, 2014; Sauter *et al.*, 2014; Sauter, 2015; Uervirojnangkoorn *et al.*, 2015; Ginn *et al.*, 2015).

2.7 Integration

Integration is the process of obtaining estimates of diffracted intensities and their standard errors from the raw images recorded on an X-ray detector (Leslie, 1999). The integration procedure can be separated into three discrete steps. The first is the determination of the reflection mask which labels pixels that are part of the reflection peak (foreground) and those in the background. The second step estimates the background values *under* the peak. Finally the peak intensity is evaluated *via* summation integration or profile fitting.

2.7.1 Background Estimation

Using the calculated profile model parameters, image pixel data are read into reflection "shoeboxes" that contain the peak pixels and a substantial border of background

pixels surrounding the peak as shown in Figure 2.7. In order to estimate the reflection intensity from the peak pixels, the background needs to be subtracted. However, since it is not possible to determine the background under the peak directly, the background in the peak region of the reflection first needs to be modelled. This is accomplished by using information from non-peak pixels in the local area of each spot. The background is typically modelled as either a constant value (Kabsch, 2010a) or a plane (Leslie, 1999) centred on the reflection. An important step in the background modelling is to ensure that the estimated background is not contaminated by outlier pixels such as zingers, unmodelled intensity from adjacent reflections, Bragg diffraction from ice, or reflections from a different lattice.

Implementation in *DIALS*

A number of different background models have been implemented in *DIALS*. The background can either be modelled independently for each image contributing to the reflection as a constant value or a plane; alternatively, the background across the whole reflection can be modelled as either a constant or a 3D hyper-plane. The constant value or plane is fit to the pixel values using a linear least-squares estimator. Additionally, *DIALS* provides a range of outlier handling methods which can be used with simple constant and linear background models and are particularly appropriate for CCD data where a pedestal has been subtracted. However, with modern photon counting detectors where the counts are Poisson distributed, these traditional methods may produce background estimates that are biased for low background levels because they assume that the pixel values are approximately normally distributed, Therefore, the default background modelling algorithm in *DIALS* uses a robust generalised linear model approach which explicitly assumes that the pixel values are Poisson distributed. This method is appropriate across the full range of observed background levels, it has been shown to be effective even when the average background is below 1 count per pixel, and is particularly suitable for photon counting detectors (Parkhurst *et al.*, 2016). This method will be described in more detail in Chapter 3.

2.7.2 Summation

Given an estimate for the background under the peak, the simplest integration algorithm is direct summation, where the integrated intensity is obtained as the sum of all background-subtracted pixel values in the peak region. Error estimates are derived from Poisson statistics as described by Leslie (1999) and summarised here. For a reflection shoebox containing m signal pixels and a background model for the signal region derived from n background pixels, the total estimated background

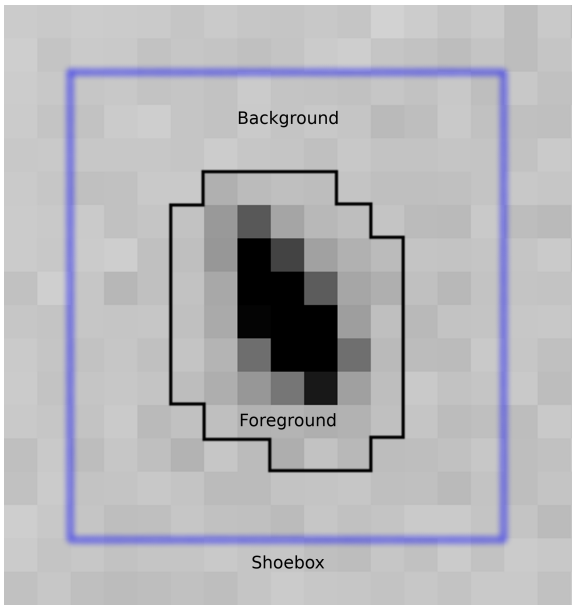


Figure 2.7: The shoebox of a reflection is the pixel aligned bounding box that contains the foreground and background pixels assigned to the reflection. A mask is used to classify the pixels as either foreground or background and this mask is generated from the profile model of the reflection shape.

counts, I_{bg}, within the signal region are

$$I_{bg} = \sum_{i}^{m} b_i. \tag{2.17}$$

The summed intensity can then be calculated simply as the sum of the observed pixel counts within the signal region, c_i, minus the total estimated background counts within the signal region, b_i, as follows

$$I_s = \sum_{i}^{m} (c_i - b_i). \tag{2.18}$$

The variance on the estimated intensity is then

$$\sigma^2_{I_s} = I_s + I_{bg} \left(1 + \frac{m}{n}\right). \tag{2.19}$$

Summation integration does not give accurate intensity estimates if a reflection's pixels are overloaded, contain otherwise invalid values, or are contaminated with signal from overlapping adjacent reflections.

Implementation in *DIALS*

A key difference between the various data processing programs is the way in which partial reflections recorded across a number of images are handled. Programs that implement 2D integration algorithms, such as *MOSFLM* (Leslie, 1999), typically

evaluate and output the intensity of each partial reflection recorded on each image independently resulting in a set of intensities for each reflection. Programs that implement 3D integration algorithms, such as *XDS* (Kabsch, 2010b) typically evaluate the total intensity of the reflection across all the images on which it was recorded and output a single intensity for each reflection. Within *DIALS*, both methods are supported and the summation intensity can be output as a single total summed intensity or a set of partial summed intensities

2.7.3 Profile fitting

One of the most important developments in the analysis of X-ray diffraction data in macromolecular crystallography was the introduction of the profile fitting method for estimating the reflection intensities. The method of profile fitting assumes that both weak and strong reflection profiles in a local region of reciprocal space all have the same normalised shape and profile (Diamond, 1969; Ford, 1974). Reference profiles can then be estimated from the measured data and applied to each reflection to provide an estimate of the reflection intensity. Profile fitting offers improvements in the estimates of intensity and standard error (Leslie, 1999) and offers additional benefits in allowing estimation of saturated reflections and in dealing with incompletely resolved diffraction spots. Since the initial exposition of the profile fitting method, three general approaches have been developed for the construction of reference reflection profiles from the measured data:

1. Empirical profile formation in detector space. Reference profiles are formed by averaging the profiles of strong reflections directly on the detector. This is commonly referred to as 2D profile fitting. This method is implemented in *MOSFLM* (Leslie, 1999) and *HKL2000/DENZO* (Otwinowski and Minor, 1997).

2. Empirical profile formation in reciprocal space. Reference profiles are formed by first transforming the reflection pixels into reciprocal space and then averaging the transformed reciprocal space profiles. This is commonly referred to as 3D profile fitting. This method is implemented in *XDS* (Kabsch, 2010b), *d*TREK* (Pflugrath, 1999) and *DIALS* (Winter *et al.*, 2018).

3. Model based profile formation. A model of the experiment is used to simulate reflection profiles. The parameters of the physical model are then modified in order to make the simulated reflection profiles match the observed profiles. This method is implemented in *EVAL15* (Schreurs *et al.*, 2009).

A brief description of the different approaches is provided here. Without exception, all popular integration programs for macromolecular crystallography contain an

implementation of the profile fitting method; programs implementing the 2D and 3D empirical profile formation methods are the most widely used.

2D profile fitting

In 2D profile fitting, reference profiles are formed by averaging the image pixels of strong reflections directly on the detector (Ford, 1974; Rossmann *et al.*, 1979; Leslie, 1999). The most widely used integration programs for macromolecular crystallography implementing 2D profile fitting algorithms are *MOSFLM* (Leslie, 1999; Leslie and Powell, 2007; Powell *et al.*, 2017) and *HKL2000/DENZO* (Otwinowski and Minor, 1997). Further discussion about 2D profile fitting will be mainly limited to the *MOSFLM* program.

Diffraction images collected using the rotation method contain a distorted view of reciprocal space. This distortion depends on various factors such as the crystal morphology and mosaicity, beam divergence and spectral dispersion, the obliquity of the diffracted beam on the detector and other details of the experimental geometry. Consequently, reflections from different parts of reciprocal space cannot be assumed to have the same recorded profile on the detector: the profile of recorded reflections changes across the face of the detector and over the course of the data collection. Within *MOSFLM*, this variation is accommodated by determining a set of reference profiles across the detector laid out in a grid formation (typically a 5x5 grid) (Leslie, 1999). In contrast, for *HKL2000/DENZO*, a reference profile is determined for each reflection by averaging the profiles of all strong reflections within a specified distance from the reflection (Otwinowski and Minor, 1997).

In *MOSFLM*, each image on which a reflection is recorded is processed separately. If a reflection is recorded over several adjacent images, each image will result in a separate measurement of the reflection intensity. In order to estimate the reflection intensity, a measurement box is positioned on each reflection such that the pixels around the predicted Bragg peak are labelled as either signal or background. The parameters defining the size of the measurement box and the signal/background mask are optimised separately for each reference profile grid point according to the procedure described by Lehmann and Larsen (1974) such that the $I/\sigma(I)$ within the measurement box is maximised. All reflections assigned to a particular grid point are then given the same measurement box.

The intensity is then evaluated by the equations given in Leslie (1999). For fully recorded reflections, the profile fitted intensity scale factor and the background plane are fitted simultaneously. For partially recorded reflections, the background plane is fitted beforehand and the profile fitted intensity scale factor is computed separately.

3D profile fitting

The quality of the profile fitted intensity estimates depends on the degree of similarity between the predicted reference profile and the observed profile of the reflections on the detector. In the rotation method, the observed reflection profiles are distorted due to various effects associated with the details of the experimental geometry. Kabsch (1988) showed that transformation of the observed reflection profiles into a reciprocal space coordinate system reduced this distortion, resulting in reflection profiles of a more uniform appearance. This enables better reference profile determination and consequently better profile fitted intensity estimates.

Kabsch (1988) uses the β-axis described by Harrison *et al.* (1985) as the basis of the transformation. In this method, the image pixels from all the images on which a reflection is predicted to be observed are used to generate a single 3D reflection profile in reciprocal space; this has come to be known as 3D profile fitting to distinguish it from 2D detector space profile fitting. Although this method is most closely associated with the *XDS* integration program (Kabsch, 2010b), it is also implemented in *d*TREK* (Pflugrath, 1999) and *DIALS* (Winter *et al.*, 2018). A discussion of the method will be provided here and the specific implementation in *DIALS* will be described in more detail later.

The algorithm is based around the local reflection specific reciprocal space coordinate system defined in Equation 2.13. The counts in each pixel of the observed reflection profile are then transformed and distributed to a grid in the reciprocal space coordinate system. As in *MOSFLM*, a set of reference profiles are formed at different points on the detector from nearby strong spots and each reflection is profile fitted to its nearest reference profile.

The 3D profile fitting method has achieved particularly widespread use with the move to more finely sliced data collection strategies. As shown in Figure 2.8, fine sliced datasets tend to have fewer fully recorded reflections, fewer spatial overlaps, lower X-ray background and fewer saturated pixels. The positional uncertainty of reflections in finely sliced data is typically smaller due to the better sampling of the individual spot shapes. Since profile fitting relies on accurate spot positions, this can result in improved profile fitted intensities. However, the benefits of fine slicing need to be balanced with the effect of detector readout noise. In the past, the use of extremely fine slicing was counter productive due to the read out noise associated with each image (Pflugrath, 1999); however, with the introduction of pixel array detectors, which have extremely low readout noise, fine slicing incurs essentially zero penalty in terms of data quality. In this regime, where almost all reflections are partially recorded within a single image, the 3D profile fitting algorithm has clear benefits over the 2D profile fitting algorithm due to its ability to use information

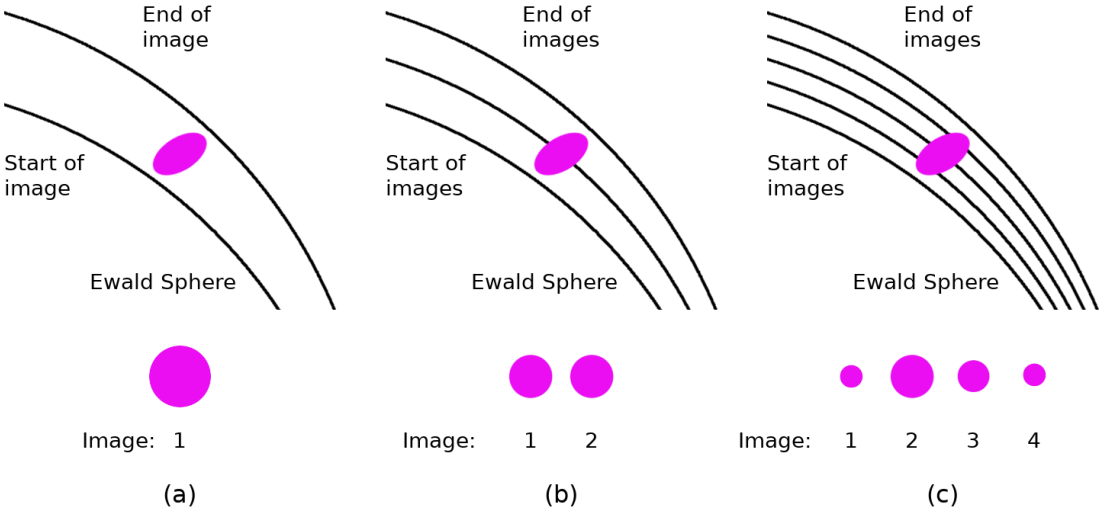


Figure 2.8: For thickly sliced data (a) many reflections will be fully recorded on a single image, for progressively finer sliced data (b) and (c) the intensity of the reflections will be spread across a greater number of images. At the bottom of each figure is an example of how the intensity from a single reflection may be distributed across multiple images.

from all images on which the reflection is recorded to perform the profile fit.

Model based profile fitting

The 2D and 3D profile fitting algorithms both rely on learning the shape of reference profiles from nearby strong spots. As such they make the assumption that the profiles of reflections in an area of reciprocal space are sufficiently similar that their profiles can be assumed to be the same. In some cases, pathologies present in the data may mean this assumption no longer holds. For example, overlapping spots, twin lattices, satellite reflections, streaky spots from modulated structures, and multi-modal spots from K-α_1/K-α_2 peak splitting may cause some profile fitting algorithms to fail or give poor results (Duisenberg *et al.*, 2003).

An alternative approach is to generate reference profiles for profile fitting from a physical model of the experiment that can produce reference profiles specific for each reflection. The idea of using a physical model to predict the profiles of Bragg reflections was first discussed in Alexander and Smith (1962) who predicted the shape of 1D profiles by considering aspects of the source size, wavelength, crystal shape and mosaicity. The *EVAL15* integration program (Schreurs *et al.*, 2009) is the most widely used program that implements a model based profile fitting algorithm. The program employs a physical model consisting of a number of parameters describing the crystal and experiment; a ray-tracing algorithm is then used to generate "general impacts" (reference profiles) which are then fit to each reflection using a least squares fit.

The benefits of this approach are that it can result in more physically realistic reference profiles, allow more accurate deconvolution of overlapping spots, and allow physical properties of the crystal itself to be determined (Schreurs *et al.*, 2009). The difficulty with such an approach is that, since the reflection profiles observed on the detector are a convolution of various effects such as crystal size and shape, mosaicity, beam divergence and dispersion and experimental geometry, an exact transformation is analytically intractable (Schreurs *et al.*, 2009). Therefore computationally intensive Monte Carlo ray tracing procedures are necessary. The difficulty then lies not in simulating the reference profiles but in determining the parameters of simulation from the data.

Benefits of profile fitting

The principal benefit of the profile fitting method is that appropriate use of the algorithm can greatly improve the standard error on the estimated intensities as compared with estimates from straight summation of the background subtracted counts (Diamond, 1969). The improvement is greatest for reflections whose intensity is small relative to the underlying background level. Leslie (1999) showed that, for very weak reflections, the use of profile fitting can reduce the standard error on the estimated intensity by a factor of $\sqrt{2}$. For strong reflections whose intensity is large relative to the underlying background level, the profile fitted intensity can be shown to converge to the summation intensity (Leslie, 1999) for an appropriate reference profile; therefore, profile fitting does not provide any benefit for strong reflections.

For a typical unimodal reference profile, the profile weight decreases with the distance from the centre of the reflection profile. As well as weighting the pixels with low counts and low signal-to-noise less strongly, this has the desirable property that, should a neighbouring reflection intrude into the signal region of the reflection being integrated, its effect on the profile fitted intensity estimate will be less deleterious than on the intensity estimate from summation integration (Leslie, 1999; Leslie, 2005). Profile fitting also permits the use of more appropriate ways of handling the issue of overlapping spots. Bourgeois *et al.* (1998) described the use of a profile fitting algorithm for the deconvolution of overlapping spots using a least-squares procedure. They reported that the intensities of spots overlapping by as much as 60% can be effectively estimated using the procedure. Profile fitting can also be used to estimate the intensity of overloaded spots by simply ignoring the contribution of the spot pixels that are overloaded; this may also be achieved using a censored regression. Intensity estimates from summation integration in this case may not be possible.

Pitfalls of profile fitting

During profile fitting, two main assumptions are made; spots close to one another in reciprocal space have the same underlying profile, and the positions of the spots can be accurately predicted (Pflugrath, 1999). Violation of either of these assumptions can cause errors in the reference profiles which in turn will cause systematic errors in the profile fitted intensities. These systematic errors will typically be small relative to the random error due to Poisson statistics for weak reflections; however, since these systematic errors tend to depend on the reflection intensity, they may be more significant for strong reflections (Leslie, 1999).

To ensure that the requirement of the first assumption is met, Kabsch (1988) developed the local reciprocal space transformation for use in the 3D profile fitting algorithm. During execution, validation of the assumption of uniform profiles can be done by computing the correlation between strong spots and their reference profiles; strong spots should have a high correlation with their reference profile (Pflugrath, 1999).

Since the reference profiles are determined from the profiles of strong reflections, the positional uncertainty in the reflection centroids will lead to broadening of the reference profiles. Even in the absence of any positional error in the reflection centroid, profile broadening will still occur due to the finite sampling size of the detector pixels. In addition to this, error in the predicted position of the reflections will also cause the reference profile to be mis-centred on the reflection. The broadening of reference profiles will cause a systematic error in all the estimated reflection intensities; however, the error in the predicted position will vary for each reflection (Leslie, 1987). It should be noted that positional errors will also affect the summation intensity estimates; if the integration shoebox is misplaced, a larger signal region will be needed which will contain more background pixels, thereby increasing the random error on the summed intensity. If the positional error is particularly large then some of the reflection signal may lie outside the integration shoebox causing a systematic error in the summed intensity.

In order to reduce the effect of positional errors in the profile fitting, it has been suggested that reflection profiles could be "auto-centred"; the position of the reflection on the detector can be varied until the optimum fit with the reference profile is found (Leslie, 1987). However, this method is typically only possible for strong reflections and may increase any systematic error in intensity estimates from profile fitting if applied to all reflections (Greenhough and Suddath, 1986; Otwinowski and Minor, 1997).

Profile fitting to partially recorded reflections can also be problematic for the 2D profile fitting algorithm. This can result in either an over- or under-estimate in

the profile fitted intensity depending on whether the reflection is more or less than 50% recorded. However, this systematic error has been shown to cancel out when the partial intensity estimates are summed to give a single fully recorded intensity estimate (Greenhough and Suddath, 1986).

Implementation in *DIALS*

In *DIALS*, 3D profile fitting is currently performed as described by Kabsch (2010a). The image/rotation-space shoebox for each reflection is first transformed into its local reciprocal space coordinate system in which the reflection profiles take on a more uniform appearance, allowing their shapes to be modelled more effectively (Kabsch, 1988). In contrast to *XDS*, the reflection data is transformed onto the reciprocal space grid by computing the overlap of each detector pixel with the transformed grid point using a polygon clipping algorithm (Sutherland and Hodgman, 1974). The fractional overlap is then used to determine the number of counts in each pixel that is distributed to each grid point in the transformed grid. An example of a reference profile is shown in Figure 2.9.

In order to aid parallel execution, blocks of images are integrated independently. The blocks of images are overlapped so that the start of a block is aligned to the centre of a preceding block. This ensures that the majority of reflections are fully recorded within a single block, with a better profile fitting intensity estimate than reflections split at block boundaries and reassembled after integration. Reference profiles are created from the strong spots at several points across the detector surface for each block of images being integrated. Each strong reflection contributes to its nearest reference profiles using a Gaussian weight derived from its distance to the reference profile such that reflections halfway between two reference profiles contribute half their intensity to each reference profile. This is done to ensure that, if there are few or no strong reflections recorded on a part of the detector, reference profiles will still be computed to allow profile fitting to be performed for weak reflections. Once the reference profiles have been created, the intensity is calculated by fitting each reflection's transformed profile to the nearest reference profile. The profile-fitted intensity and error are calculated as described by Kabsch (2010a).

2.7.4 Data correction

After integration, a number of corrections are applied to the raw intensity estimates. As a crystal is rotated about a fixed axis, different reflections will be in the diffracting condition for different lengths of time depending on their position in reciprocal space; this results in a predictable difference in observed intensity which is corrected for by the Lorentz correction (Zachariasen, 1945) which is calculated as follows:

Figure 2.9: A reference profile used in profile fitting. Each image corresponds to a slice through the 3D profile.

$$L = \frac{|\boldsymbol{s}_1 \cdot (\boldsymbol{m}_2 \times \boldsymbol{s}_0)|}{|\boldsymbol{s}_1||\boldsymbol{s}_0|}. \tag{2.20}$$

The scattering intensity is also dependent on the polarisation of the incident X-ray beam. Given a polarisation normal, $\boldsymbol{n}_p$, and the polarisation fraction, f_p, the polarisation correction (Zachariasen, 1945) is given by

$$P = (1 - 2f_p)\left(1 - \left(\frac{\boldsymbol{n}_p \cdot \boldsymbol{s}_1}{|\boldsymbol{s}_1|}\right)^2\right) + f_p\left(1 + \left(\frac{\boldsymbol{s}_1 \cdot \boldsymbol{s}_0}{|\boldsymbol{s}_1||\boldsymbol{s}_0|}\right)^2\right). \tag{2.21}$$

For pixel array detectors, the thickness of the detector pixels also has an effect on the reflection intensities which requires correcting. Each X-ray photon has a probability of being absorbed by the sensor after travelling a distance, t, through the sensor. The effective thickness of the detector depends on the angle of incidence of the diffracted beam vector, θ and the probability of it being absorbed depends on the wavelength (and material) dependent X-ray linear attenuation coefficient, μ_a, of the sensor (NIST, 2004). The correction, known as the detector quantum efficiency (Winter *et al.*, 2018) is given by

$$Q = 1 - \exp\left(-\mu_a(\lambda)\frac{t}{\cos(\theta)}\right). \tag{2.22}$$

The corrected intensity is then $I_{corr} = I(L/P)(1/Q)$.

2.8 Data reduction

Once integration has provided individual intensity estimates for all observed reflections in a dataset, data reduction is then used to compute a merged average intensity estimate for each symmetry equivalent reflection. This requires the point group to be specified; however, since it is generally not possible to determine the true space group until the structure has been solved, a likely space group may be used in the first instance. The first step of data reduction is, therefore, to select a probable space group by analysing the unit cell and reflection intensities. Within the *CCP4* suite, this task is performed by the *POINTLESS* program (Evans, 2005).

Once the probable space group has been determined, symmetry equivalent reflections need to be re-scaled before they can then be merged; This is because various physical experimental factors result in the individual intensities being recorded on different scales. These physical factors include those affecting the incident beam, such as fluctuations in intensity and absorption of the incident beam by the crystal. There are also factors related to the rotation of the crystal about the fixed rotation axis, such as changes in the illuminated crystal volume, absorption of the diffracted beam and radiation damage. Additionally, there may be factors associated with the

detector, such as variations in sensitivity across the detector surface, shadowing on the detector and, in older detectors, shutter synchronisation errors (Evans, 2005; Otwinowski *et al.*, 2003; Kabsch, 2010a). To merge the symmetry equivalent intensity estimates, a scale factor needs to be applied to each intensity estimate to correct for these physical factors and to ensure that all intensities are on a common scale. Within the *CCP4* suite this task is performed by the *AIMLESS* program (Evans and Murshudov, 2013). The parameters of the model are determined by minimising the differences between symmetry equivalent observations to make the data as internally consistent as possible according to the the following equation (Hamilton *et al.*, 1965; Fox and Holmes, 1966; Ford and Rollett, 1968):

$$\Psi = \sum_{h} \sum_{l} w_{hl}(I_{hl} - g_{hl} < I_h >)^2. \tag{2.23}$$

Where, h is an index over all unique reflections, l is an index over all measurements of a particular unique reflection, w_{hl} is the weight given to a particular observation (typically the inverse of the variance), I_{hl} is an observed intensity, g_{hl} is the inverse scale factor on the intensity, and $< I_h >$ is the true merged intensity. The merged intensities are then calculated simply as:

$$< I_h >= \frac{\sum_{l} w_{hl} g_{hl} I_{hl}}{\sum_{l} w_{hl} g_{hl}^2}. \tag{2.24}$$

.

The errors on the integrated intensities are typically underestimated since they are usually derived from Poisson statistics and do not take into account other potential sources of error such as errors in the predicted position and errors resulting from profile fitting. Therefore, data reduction programs attempt to improve the error estimates by applying a correction to ensure that the normalised deviations from the mean intensity, $\delta_{hl} = (I_{hl} - < I_h >)/\sigma(I_{hl})$, follow a standard normal distribution (Evans, 2005). This almost always results in an increase in the estimated errors on the reflection intensities.

Data reduction programs differ in the implementation of their scaling model. A simple scaling model, might apply a scale factor per image (known as batch scaling); however, this model leads to discontinuities in the scale factors between adjacent images. Therefore, a smoothly varying model is often used and may perform better. A smooth model may be implemented by determining the scale factors at regular intervals and interpolating using Gaussian weights as done in *AIMLESS* (Evans and Murshudov, 2013). Scaling models also incorporate a B-factor term which provides a resolution dependent radiation damage correction which is allowed to vary as a function of time (or rotation). The B-factor correction is given

by $\exp(-2B\sin^2(\theta)/\lambda^2)$. *SCALEPACK* (Otwinowski *et al.*, 2003) implements an alternative scaling model using exponential modelling of the scaling parameters which are estimated *via* the minimisation of a chi-squared like target function. *XSCALE* (Kabsch, 2010a) implements a non-parametric rather than physical scaling model in which a finite grid of correction factors is used.

Once data reduction is completed, it is necessary to determine the structure factor amplitudes from the merged intensities. This involves computing the square root of each merged intensity. Problems result if the merged intensity is negative as this would yield an imaginary structure factor amplitude. Excluding negative reflections, or setting them to zero, biases the distribution of intensities resulting in a perturbation of the electron density map and refined atomic model (French and Wilson, 2012). In order to avoid this problem, programs such as *CTRUNCATE* (Winn *et al.*, 2011) implement a procedure using Bayesian statistics to ensure that all the intensity estimates are positive, with the Wilson distribution (Wilson, 1949) being used as a prior distribution for the reflection intensities (French and Wilson, 1978). Application of this model results in negative and small positive intensities being modified but does not result in much change in large positive intensities. These corrected intensities are then used to determine the structure factor amplitudes. It should be noted that these corrected intensities should not be used in any further analysis since they are biased. Indeed, rather than using structure factor amplitudes, modern phasing and refinement programs are moving towards using the intensities themselves to avoid explicitly performing this correction.

Chapter 3

Robust background modelling using Generalised Linear Models

3.1 Introduction

In macro-molecular crystallography (MX), integration programs - such as *MOSFLM* (Leslie, 1999), *XDS* (Kabsch, 2010b), *d*TREK* (Pflugrath, 1999) and *DIALS* (Waterman *et al.*, 2013; Winter *et al.*, 2018) - are used to estimate the intensities of individual Bragg reflections from a set of X-ray diffraction images. Whilst details of the processing differ, these programs all follow the same basic procedure to calculate the intensity estimates. For each reflection, pixels in the neighbourhood of the predicted Bragg peak are labelled as either "foreground" or "background" pixels through the application of a model of the shape of the reflection on the detector. The reflection intensity may be estimated by subtracting the sum of the estimated background values from the sum of the total number of counts in the foreground region. This is termed "summation integration". The background in the foreground region is unknown and is, therefore, estimated from the surrounding background pixels assuming smooth variation in the background counts.

An accurate estimate of the background is a prerequisite for deriving an accurate estimate of the reflection intensity. Integration programs typically assume that the background in the vicinity of a reflection peak can either be modelled as a constant value (Kabsch, 2010a) or a plane with a small gradient (Leslie, 1999). Since the reflection peak typically extends across an area containing a small number of pixels, these assumptions generally hold true and the resulting simple models have the advantage of being computationally inexpensive to calculate from the surrounding background pixels.

The situation is complicated by the presence of pixels whose values appear not to be drawn from the same distribution as other pixels in the background region

assuming the simple background model. Typically these pixels contain a higher number of counts relative to their neighbours than would be expected if they were drawn from the same distribution. For example, the counts can be the result of hot pixels (defective pixels which always show a large number of counts), zingers (random unmodelled spikes in intensity from, for example, cosmic rays), intensity from adjacent reflections, ice rings, or other unmodelled intensity. Background estimation routines in integration programs need to be resistant to such outlier pixels. Therefore programs implement methods to exclude outliers from the background calculation.

In this chapter, a new outlier handling method using robust generalised linear models is introduced. This algorithm is implemented within the *DIALS* framework which also provides implementations of several simple outlier handling algorithms as well as algorithms used in other integration packages. The application of this new algorithm is then compared with the application of the other outlier handling methods available within the *DIALS* framework. The following methods have been implemented in *DIALS*:

1. *null*. No outlier handling is used.

2. *truncated*. This method excludes extreme pixel values by discarding a fraction of the the pixels (by default 5%) containing the highest and lowest number of counts.

3. *nsigma*. This method excludes extreme pixel values by computing the mean and standard deviation (σ) of the pixel values and computing a threshold such that all pixels with values outside $\mu \pm N\sigma$ are discarded, where the default value for parameter N is 3. In our implementation, the procedure is applied once; however, an alternative approach may be to apply the procedure iteratively.

4. *tukey*. This method excludes extreme pixel values by computing the median and interquartile range (IQR). Pixels with values $< Q_1 - N * \text{IQR}$ and values $> Q_3 + N * \text{IQR}$ are discarded, where Q_1 and Q_3 are the first and third quartile respectively and the default value for N is 1.5

5. *plane*. This is an implementation of the method used in *MOSFLM* (Leslie, 1999). The authors were fortunate to have access to the *MOSFLM* source code and were, therefore, able to verify that the algorithm implemented in DIALS gave equivalent results. First a percentage of the highest valued pixels are discarded and a plane is computed from the remaining background pixels such that the modelled background at each pixel position (x, y) is $z = a + bx + cy$, where the origin of x and y is at the peak position. The value of a is, therefore,

the mean background. All pixels are then checked and those with an absolute deviation from the plane $|z_{obs} - z| > N\sqrt{a}$, where the default value for N is 4 are discarded. The plane could then be iteratively refitted; however, within the implementation in *DIALS*, the plane is only fit once.

6. *normal*. This is an implementation of the method described in Kabsch (2010a). The method assumes that the pixel values in the background region are approximated by a normal distribution. The pixels are sorted by increasing value and their distribution checked for normality. The highest valued pixels are then iteratively removed until the distribution of the remaining pixels is approximately normal. It should be noted that the authors did not have access to the *XDS* source code so were unable to verify that the algorithm implemented in DIALS gave equivalent results. Additionally, newer versions of *XDS* adapted for low background data use a different method (Diederichs, 2015).

7. *glm*. The robust generalised linear model algorithm described in this chapter.

Most of the methods for handling outliers described above rely on the assumption that the pixel values are drawn from a normal distribution, whereas in reality the data are Poisson distributed. As the mean expected value increases, a Poisson distribution is well approximated by a normal distribution; however, as the mean tends towards zero, this approximation becomes increasingly inappropriate. Therefore, although successfully used for data collected on CCD detectors, traditional methods may have problems when used on data collected on photon counting detectors such as the DECTRIS PILATUS (Henrich *et al.*, 2009). Data collected using these detectors are associated with having a lower background than data collected on CCD detectors. This is partly due to the opportunity for collecting increasingly fine ϕ-sliced data offered by these detectors due to the fast readout and reduced noise associated with them (Mueller *et al.*, 2011). Additionally, new beamlines have been designed where the whole experiment, including the sample and detector, is kept under vacuum (Wagner *et al.*, 2016) yielding data with very low background due to the absence of scattering by the air. Furthermore, the design of beamlines has also contributed to the ability to collect data with lower background. Evans *et al.* (2011) showed how, for small crystals, matching the beam size to the size of the crystal could result in a drastic reduction in the X-ray background by reducing the volume of non-diffracting material that the X-rays impinge upon.

Intuitively, outlier handling methods which remove values purely from one side of the distribution will result in a biased estimate of the Poisson mean. Since the Poisson distribution is asymmetric, simple outlier handling techniques which remove

a fixed percentage of pixels from either side (as in the *truncated* method described above) may also introduce bias. The bias for the truncated estimator of the Poisson mean is given below:

$$\begin{aligned}\lambda - E[\lambda_{trunc}] &= \lambda - \frac{\sum_{j=a}^{b} jP(y=j)}{\sum_{j=a}^{b} P(y=j)} \\ &= \lambda\left(1 - \frac{Q(b,\lambda) - Q(a-1,\lambda)}{Q(b+1,\lambda) - Q(a,\lambda)}\right).\end{aligned} \tag{3.1}$$

Here $E[\lambda_{trunc}]$ is the expected value of the truncated estimator and $Q(x,\lambda) = \Gamma(x,\lambda)/\Gamma(x)$ is the regularised gamma function. The bias of the estimator is dependent on the Poisson mean λ. In the case of the non-truncated estimate of the mean of a Poisson distribution, $a = 0$ and $b = \infty$. $Q(\infty,\lambda) = 1$ and $Q(0,\lambda) = Q(-1,\lambda) = 0$; therefore, the bias of the non-truncated estimator is zero. It should be noted that any method which attempts to remove outliers from the data will systematically reduce the variance of the distribution even when no outliers are present.

In this chapter, it is shown how the use of inappropriate outlier handling methods can lead to poor background determination and systematic bias in the estimated background level. The use of a simple robust estimation method using generalised linear models where the pixel values are explicitly assumed to be drawn from a Poisson distribution is proposed. It is shown that use of this algorithm results in superior statistical behaviour compared to existing algorithms.

3.2 Algorithm

3.2.1 Generalised linear models

Generalised linear models, first described in Nelder and Wedderburn (1972), are a generalisation of ordinary linear regression. In linear regression, the errors in the dependent variables are assumed to be normally distributed. Generalised linear models extend this to allow the errors in the dependent variables to be drawn from a range of distributions in the over-dispersed exponential family, including the Poisson distribution.

The exponential family of probability distributions

The exponential family contains many well known probability distributions, including the Normal, Poisson, Binomial and Gamma distributions; the general form is defined

as follows:

$$f(x|\theta) = h(x)\exp\left(\eta(\theta)T(x) - A(\eta)\right). \tag{3.2}$$

Where, x is the random variable and θ is a parameter of the distribution; $h(x)$, $\eta(\theta)$, $T(x)$ and $A(\eta)$ are known functions and in the case of the Poisson distribution are given by:

$$h(x) = \frac{1}{x!}, \quad \eta = \log(\theta), \quad T(x) = x, \quad A(\eta) = e^{\eta}. \tag{3.3}$$

Link function

In linear regression, the dependent variables, μ, are related to the matrix of explanatory variables (also known as the design matrix), $\boldsymbol{X}$, and the vector of model parameters, $\boldsymbol{\beta}$, *via* a linear predictor, such that $\mu = \eta = \boldsymbol{X\beta}$. In the generalised linear model framework, the linear predictor, $\eta = \boldsymbol{X\beta}$, is related to the dependent variables *via* a link function, $g(\mu) = \eta$, which depends on the probability distribution. For the Poisson model, the link function is the natural logarithm, so that $\log(\mu) = \eta = \boldsymbol{X\beta}$; therefore, the expected values are given by, $\mu = e^{\boldsymbol{X\beta}}$. The generalised linear model framework essentially allows the expected value of the response to be transformed, thereby avoiding any need to transform the data itself. The maximum likelihood estimate is typically found using iteratively reweighted least squares. This is done as it is computationally flexible and allows a numerical solution to be found when it is difficult to maximise the likelihood function directly.

3.2.2 Robust estimation

A method to apply robust estimation to the generalised linear model framework is described by Cantoni and Ronchetti (2001). For convenience, the terms used in the following equations are given in the list of symbols on page 13. The maximum likelihood function is replaced by a quasi-likelihood estimator whose score function, $\boldsymbol{U}$ is given by:

$$\boldsymbol{U} = \sum_{i=1}^{n} \left[\psi_c(r_i) w(\boldsymbol{x_i}) \frac{{\mu_i}'}{\sqrt{\phi v_{\mu_i}}} - a(\boldsymbol{\beta}) \right] = 0. \tag{3.4}$$

Here $\boldsymbol{x_i}$ is a row of the design matrix, ${\mu_i}' = \frac{\partial \mu_i}{\partial \boldsymbol{\beta}} = \frac{\partial \mu_i}{\partial \eta_i}\boldsymbol{x_i}$, and $r_i = \frac{y_i - \mu_i}{\sqrt{v_{\mu_i}}}$ are the Pearson residuals for each observation, y_i, with respect to its expected value μ_i and variance v_{μ_i}. ϕ is the dispersion, which, for a Poisson distribution is known to be equal to 1. The functions $w(\boldsymbol{x_i})$ and $\psi_c(r_i)$ provide weights on the explanatory variables and dependent variables respectively. Here, since it is assumed that each

pixel has identical weighting, the weights for the explanatory variables, x, are set to 1 (*i.e.* $w(\boldsymbol{x_i}) = 1$). The weighting on the dependent variables, $\psi_c(r_i)$, gives the estimator its robust characteristics. In this application of the algorithm, the Huber weighting function is used as described by Cantoni and Ronchetti (2001) and shown below:

$$\psi_c(r_i) = \begin{cases} r_i, & |r_i| \leq c \\ c * \text{sgn}(r_i), & |r_i| > c \end{cases}. \tag{3.5}$$

This weighting function has the effect of damping values outside a range defined by the tuning constant, c. A value of $c = 1.345$ is used, corresponding to an efficiency of 95% for a normal distribution (Heritier *et al.*, 2009). The efficiency of an estimator is a measure of how optimal the estimator is relative to the best possible estimator. Increasing the value of the tuning parameter increases the efficiency of the estimator but decreases its robustness to outliers. A value of $c = \infty$ results in the same estimator as for the standard GLM approach.

The constant $a(\boldsymbol{\beta})$ is a correction term used to ensure Fisher consistency; *i.e.* the correction term ensures that an estimate based on the entire population, rather than a finite sample, would result in the true parameter value being obtained (Fisher, 1922). The estimator is said to be Fisher consistent if $E[\boldsymbol{U}] = 0$. The correction term is computed simply by expanding $E[\boldsymbol{U}] = \sum_{i=1}^{n} \left[E[\psi_c(r_i)] w(\boldsymbol{x_i}) \frac{\mu_i'}{\sqrt{v_{\mu_i}}} - a(\boldsymbol{\beta}) \right] = 0$ and is given by:

$$a(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} E[\psi_c(r_i)] w(\boldsymbol{x_i}) \frac{\mu_i'}{\sqrt{v_{\mu_i}}}. \tag{3.6}$$

The algorithm was implemented in C++ for use within *DIALS*. It is available in the *glmtbx* package within the *cctbx* library (Grosse-Kunstleve *et al.*, 2002). In this implementation, the parameter estimates are obtained by solving equation 3.4 using iteratively reweighted least squares as described in Cantoni and Ronchetti (2001) and Heritier *et al.* (2009).

3.2.3 Robust GLM algorithm implementation in *DIALS*

The equations for asymptotic variance of the estimator in Cantoni and Ronchetti, 2001, Appendix B and Heritier *et al.*, 2009, Appendix E.2 contain an error (Cantoni 2015 private communication). A description of the algorithm, including corrections, is given here.

The background, μ_i, at each pixel is estimated from the generalised linear model as $\log(\mu_i) = \boldsymbol{X\beta}$. Given initial model parameter estimates $\boldsymbol{\beta}^{(t)}$, the parameter estimate for the next iteration of the algorithm, $t + 1$, is given by:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathcal{I}^{-1}\boldsymbol{U}. \tag{3.7}$$

Where $\mathcal{I}$ is the Fisher information matrix and $\boldsymbol{U}$ is the scoring function, given by:

$$\begin{aligned} \boldsymbol{U} &= \sum_{i=1}^{n} \left[\psi_c(r_i) w(\boldsymbol{x_i}) \frac{{\mu_i}'}{\sqrt{v_{\mu_i}}} - a(\boldsymbol{\beta}) \right] \\ &= \sum_{i=1}^{n} \left[(\psi_c(r_i) - E[\psi_c(r_i)]) w(\boldsymbol{x_i}) \frac{{\mu_i}'}{\sqrt{v_{\mu_i}}}) \right]. \end{aligned} \tag{3.8}$$

The only additional term that needs to be calculated here is the expectation $E[\psi_c(r_i)]$. In order to compute this, let us denote $j_1 = \lfloor \mu_i - c\sqrt{\phi v_{\mu_i}} \rfloor$ and $j_2 = \lfloor \mu_i + c\sqrt{\phi v_{\mu_i}} \rfloor$. For a Poisson distribution

$$\sum_{a}^{b} \left(\frac{j}{\mu} - 1 \right) P(y = j) = P(y = a - 1) - P(y = b). \tag{3.9}$$

The expectation, $E[\psi_c(r_i)]$, is then given by:

$$\begin{aligned} E[\psi_c(r_i)] &= \sum_{j=0}^{\infty} \psi_c \left(\frac{j - \mu_i}{\sqrt{v_{\mu_i}}} \right) P(y_i = j) \\ &= c(P(y_i \geq j_2 + 1) - P(y_i \leq j_1)) \\ &+ \frac{\mu_i}{\sqrt{v_{\mu_i}}} (P(y_i = j_1) - P(y_i = j_2)). \end{aligned} \tag{3.10}$$

The Fisher information matrix, $\mathcal{I}$, is given by:

$$\mathcal{I} = E \left[-\frac{\partial \boldsymbol{U}}{\partial \boldsymbol{\beta}} \bigg|_{\boldsymbol{\beta} = \boldsymbol{\beta}^{(t)}} \right] = \boldsymbol{X}^T \boldsymbol{B} \boldsymbol{X}. \tag{3.11}$$

The diagonal components of the matrix $\boldsymbol{B}$ are given by:

$$b_i = E \left[\psi_c(r_i) \frac{\partial}{\partial \mu_i} \log(P(y_i | x_i, \mu_i)) \right] \frac{w(\boldsymbol{x_i}) \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2}{\sqrt{v_{\mu_i}}}. \tag{3.12}$$

For a Poisson distribution, $\frac{\partial \mu_i}{\partial \eta_i} = \frac{\partial e^{\eta_i}}{\partial \eta_i} = e^{\eta_i} = \mu_i$ and $\frac{\partial}{\partial \mu_i} \log(P(y_i | x_i, \mu_i)) = \frac{y_i - \mu_i}{\mu_i} = \frac{y_i - \mu_i}{v_{\mu_i}}$. The expectation is given by:

$$
\begin{aligned}
E\left[\psi_c(r_i)\frac{\partial}{\partial \mu_i}\log(P(y_i|x_i,\mu_i))\right] &= E\left[\psi_c\left(\frac{y_i-\mu_i}{\sqrt{v_{\mu_i}}}\right)\frac{y_i-\mu_i}{v_{\mu_i}}\right] \\
&= \sum_{j=0}^{\infty}\psi_c\left(\frac{j-\mu_i}{\sqrt{v_{\mu_i}}}\right)\frac{j-\mu_i}{v_{\mu_i}}P(y_i=j) \\
&= c\frac{\mu_i}{v_{\mu_i}}(P(y_i=j_1)+P(y_i=j_2)) \\
&+ \frac{{\mu_i}^2}{{v_{\mu_i}}^{3/2}}(P(y_i=j_1-1)-P(y_i=j_2-1) \\
&+ \frac{1}{\mu_i}P(j_1<=y_i<=j_2-1)-P(y_i=j_1)+P(y_i=j_2)).
\end{aligned}
\tag{3.13}
$$

3.2.4 Background models

In applying the GLM approach to modelling of the background pixel intensity, instead of modelling the expected background as a constant or a plane, the logarithm of the expected background is modelled as a constant or a plane. It should be noted that, for a constant background model, the two are equivalent. The rows of the design matrix for the constant and planar model are $\boldsymbol{x_i} = (1)$ and $\boldsymbol{x_i} = (1, p_i, q_i)$ respectively, where (p_i, q_i) is the coordinate of the ith pixel on the detector.

Considering that the algorithm will be applied to each reflection in the dataset independently, and a typical X-ray diffraction dataset contains many reflections (a high multiplicity dataset may have $> 10^6$ reflections), there is a requirement for the algorithm to be computationally efficient. Since the background models being used are very simple, the general algorithm can be simplified; this is described below for the constant background model.

3.2.5 Simplified algorithm for constant background model

In the case of the constant background model (*i.e.* where a robust estimate of the mean of the background pixels is calculated), the model only has a single parameter, β, and the rows of the design matrix, $\boldsymbol{X}$, are all defined as $x_i = 1$. The estimate of the background is then, $\mu_i = \mu = \exp(\beta)$ and the iterative algorithm to estimate the model parameter, β is simplified to the following:

$$
\beta^{(t+1)} = \beta^{(t)} + \boldsymbol{U}/\mathcal{I}. \tag{3.14}
$$

Since the expectations of $E[\psi_c(r_i)]$ and $E\left[\psi_c(r_i)\frac{\partial}{\partial \mu}\log(P(y_i|x_i,\mu))\right]$ do not depend on y_i, and $\mu_i = \mu$ is the same for each point, they are constant for a given value of μ as shown below:

$$
\begin{aligned}
C_1(\mu) &= E[\psi_c(r_i)] \\
&= c(P(y_i \geq j_2 + 1) - P(y_i \leq j_1)) \\
&+ \sqrt{\mu}(P(y_i = j_1) - P(y_i = j_2)).
\end{aligned} \tag{3.15}
$$

$$
\begin{aligned}
C_2(\mu) &= E\left[\psi_c(r_i)\frac{\partial}{\partial \mu}\log(P(y_i|x,\mu))\right] \\
&= c(P(y_i = j_1) + P(y_i = j_2)) \\
&+ \sqrt{\mu}(P(y_i = j_1 - 1) - P(y_i = j_2 - 1) \\
&+ \frac{1}{\mu}P(j_1 <= y_i <= j_2 - 1) - P(y_i = j_1) + P(y_i = j_2)).
\end{aligned} \tag{3.16}
$$

The scoring function, $\boldsymbol{U}$ and the Fisher information, $\mathcal{I}$ are then simplified to the following:

$$
\boldsymbol{U} = \left(\sum_{i=1}^{n} \psi_c(r_i) - nC_1(\mu)\right)\sqrt{\mu} \tag{3.17}
$$

$$
\mathcal{I} = nC_2(\mu)\mu\sqrt{\mu}. \tag{3.18}
$$

The updated value of the parameter estimate, $\beta^{(t+1)}$ is then given by:

$$
\beta^{(t+1)} = \beta^{(t)} + \frac{\sum_{i=1}^{n} \psi_c(r_i) - nC_1(\mu)}{n\mu C_2(\mu)}. \tag{3.19}
$$

3.3 Analysis

3.3.1 Experimental data

To evaluate how different outlier handling methods affect the quality of the processed data, four datasets were used. For each dataset, the average background pixel value, binned by resolution can be seen in Figure 3.1 and a randomly selected spot, observed at 3Å, is shown in Figure 3.2. In each case, the background is primarily composed of pixels with 0 or 1 counts in them. Any algorithm which assumes a normal distribution of pixel values is expected to perform badly on this data.

1. A weak Thaumatin dataset collected on Diamond beamline I04 and available online (Winter and Hall, 2014). This dataset was chosen as it is a standard test dataset used by the *DIALS* development team. The average background over all resolution ranges is less than 1 count per pixel and there is also a low

incidence of outliers in the background pixels; an outlier handling algorithm should be able to handle good data without degrading it. The dataset was processed to a resolution of 1.2 Å.

2. A Ruthenium Polypyridyl complex bound to duplex DNA (Hall *et al.*, 2011) collected at Diamond beamline I02 and available online (Winter and Hall, 2016). This dataset was chosen due to the presence of a number of outliers in the background that were observed to cause the wrong point group to be found in the downstream data processing. The dataset was processed to a resolution of 1.2 Å. The average background is around 2.5 counts per pixel at low resolution but decreases rapidly at high resolution.

3. A weak Thermolysin dataset collected on Diamond beamline I03 and available online (Winter and McAuley, 2016). This dataset was chosen because it is extremely weak, with an average background of less than 0.15 counts per pixel across the whole resolution range. Additionally, it was observed that some data processing programs gave poor results for this data, which was attributed to the poor handling of the low background. The dataset was processed to a resolution of 1.5 Å.

4. A weak FutA dataset collected on Diamond beamline VMX-m. This dataset was collected during one of the first user data collections on the beamline. The dataset was chosen as an example of the type of data that will be typical of this beamline; the reflections are very weak and the background is less than 0.15 counts per pixel across the entire resolution range. The dataset consists of a number of "wedges" covering a small rotation; out of an initial 28 wedges, 17 were successfully indexed and integrated. Subsequently, 12 wedges were scaled and merged together to give the dataset used here. The dataset was processed to a resolution of 2.1 Å.

3.3.2 Data analysis

Each dataset was processed with *xia2* (Winter, 2009) using *DIALS* (Winter *et al.*, 2018) as the data analysis engine. Subsequent data reduction was performed in *xia2* using the programs *POINTLESS* (Evans, 2005), *AIMLESS* (Evans and Murshudov, 2013) and *CTRUNCATE* (Winn *et al.*, 2011). Identical data processing protocols were used for each dataset with the exception of the choice of outlier handling method. Details of how the data processing was performed are given in Section A.1

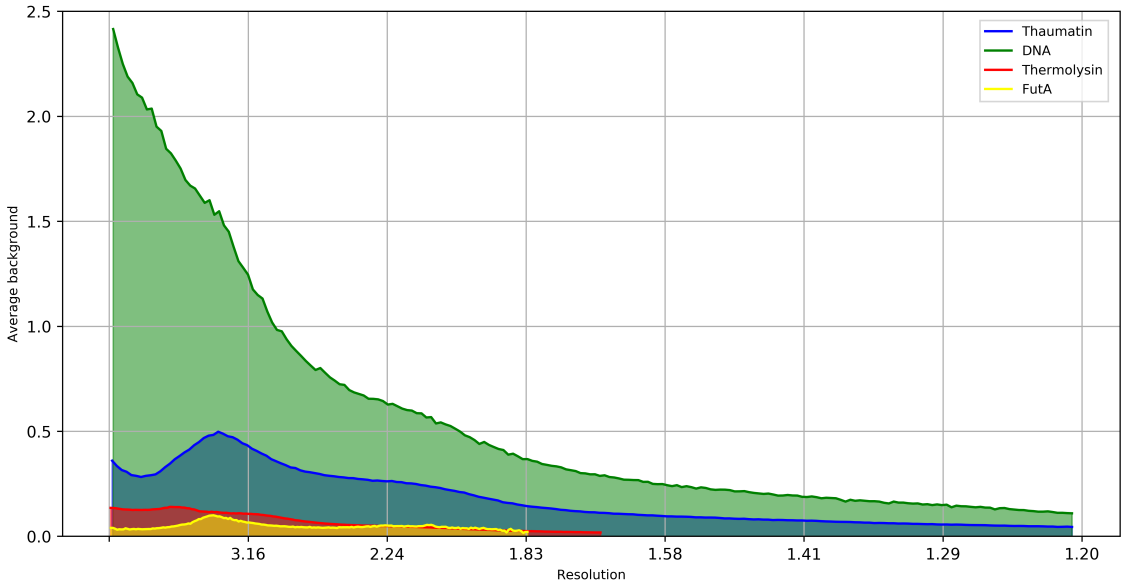


Figure 3.1: The average background level across the resolution range for each dataset.

3.3.3 Background estimates

In general, for well-measured data, pixel outliers in the background region should only affect a minority of reflections. This is the case for the four datasets considered here, where most reflections are free from pixel outliers in the background region. It is therefore expected that, for the majority of reflections, the background estimates using a well behaved outlier handling algorithm should be comparable to those using no outlier handling. Figure 3.3 shows a histogram of the normalised difference in background estimates with and without outlier handling for five existing methods and the GLM approach adopted here.

It can be seen that the traditional outlier handling methods introduce negative bias into the background estimate; the background level is systematically lower than that using no outlier handling. Of additional concern is the percentage of reflections whose background is estimated as exactly zero due to all non-zero valued pixels in the background being rejected by the outlier handling algorithm, as shown in Table 3.1. For some of the algorithms, this percentage is very high, particularly when applied to the very weak Themolysin and FutA datasets, indicating that for low background levels, the algorithm is rejecting all non-zero pixels as outliers. In contrast, for the GLM method, it can be readily seen that the background estimates show significantly less systematic bias in the background level than seen for the other methods. On average, the background estimates resulting from the GLM method are approximately equal to those with no outlier handling. The mean normalised difference between the estimates from the GLM method and the estimates with no outlier handling are -3.67×10^{-5}, -8.38×10^{-4}, 3.38×10^{-4} and -2.93^{-3} for the

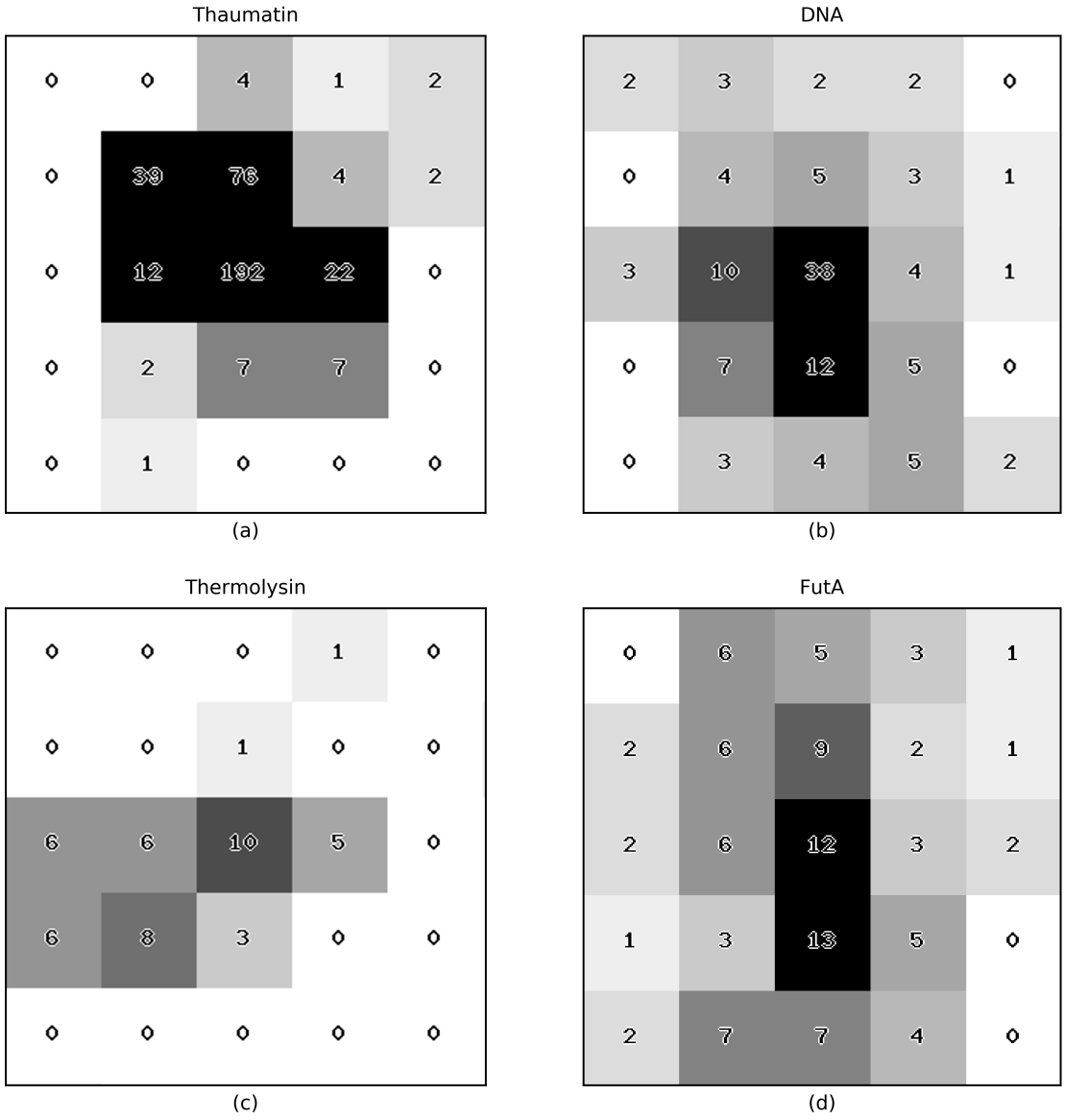


Figure 3.2: An example reflection shoebox with pixel values, observed at 3Å, for (a) Thaumatin, (b) DNA, (c) Thermolysin and (d) FutA.

Table 3.1: The percentage of reflections where all non-zero pixels were rejected by the outlier handling algorithm resulting in a background estimate of zero counts per pixel.

	Thaumatin	DNA	Thermolysin	FutA
truncated	0.0%	0.0%	0.0%	2.7%
nsigma	31.3%	0.9%	76.3%	88.5%
tukey	77.9%	56.8%	95.0%	98.5%
plane	0.7%	0.0%	30.2%	32.7%
normal	37.0%	0.0%	78.2%	88.9%
glm	0.0%	0.0%	0.0%	0.0%

Thaumatin, DNA, Thermolysin and FutA datasets respectively.

To test the behaviour of the GLM method in the presence of outlier pixels, we selected Bragg reflections whose background regions contained outliers as follows. Reflections whose background pixels have an index of dispersion (variance/mean) > 10 were selected and on this basis 15 out of 389442 reflections were chosen for the Thaumatin dataset, 60 of out 219431 for the DNA dataset, 272 out of 3322808 for the Thermolysin dataset and 53 out of 39606 for the FutA dataset. For Poisson distributed data, the index of dispersion should be equal to 1 (with a variance of $2/(N-1)$, where N is the sample size). Values much greater than 1 indicate that the pixel values are over-dispersed relative to a Poisson distribution. This indicates that the pixel values are not all drawn from the same distribution and thus provides a reasonable, straightforward, method of selecting reflections with potential pixel outliers.

Figure 3.4 shows the difference between the estimated background and the median background value (*i.e.* the most robust estimate of the background) for no outlier handling and for the GLM method. It should be noted that whilst the median is the most robust estimate, in the sense that it is the estimate of central tendency least susceptible to outliers, it is not appropriate for use here because, for very low background, the median is likely to be equal to zero and the background will be systematically underestimated. However, for a Poisson distribution with rate parameter λ, the bounds of the median are $\lambda - \ln(2) <= \text{median} < \lambda + 1/3$ (Choi, 1994); a robust estimate of the background level should be within these bounds. As expected, with no outlier handling, the background estimate is vastly overestimated for increasing index of dispersion. With the robust GLM algorithm, the estimated background value is within the bounds given by the median background value, indicating that the algorithm is adequately handling outliers.

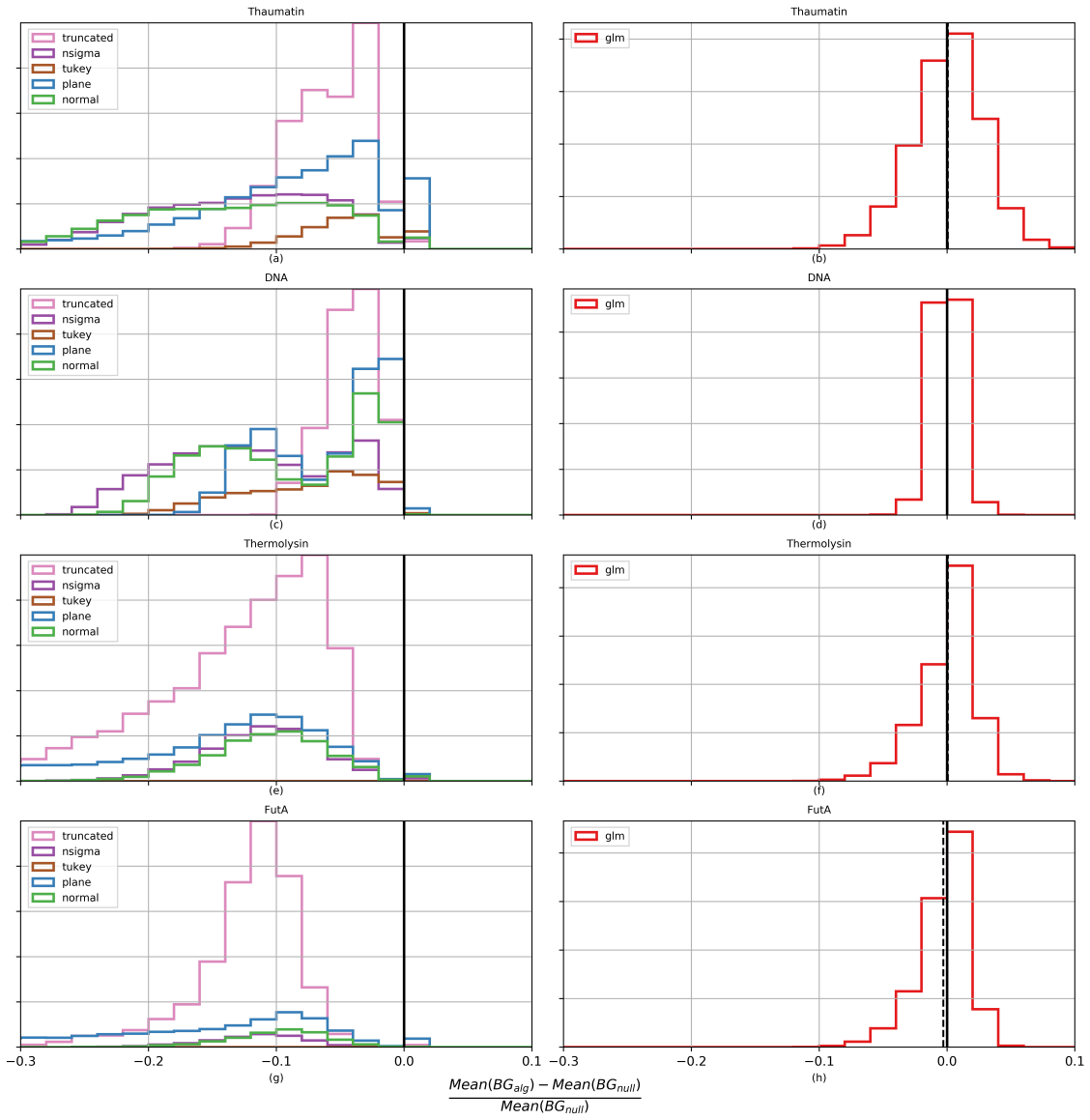


Figure 3.3: Histogram of normalised differences between the mean background with outlier handling for each outlier algorithm and the mean background with no outlier handling. For clarity, the plots for the GLM method are shown separately. The vertical black line indicates zero difference between the estimates. The estimates using the GLM algorithm are distributed more symmetrically around the null estimates, while all the other algorithms show significant systematic bias in the estimated background levels.

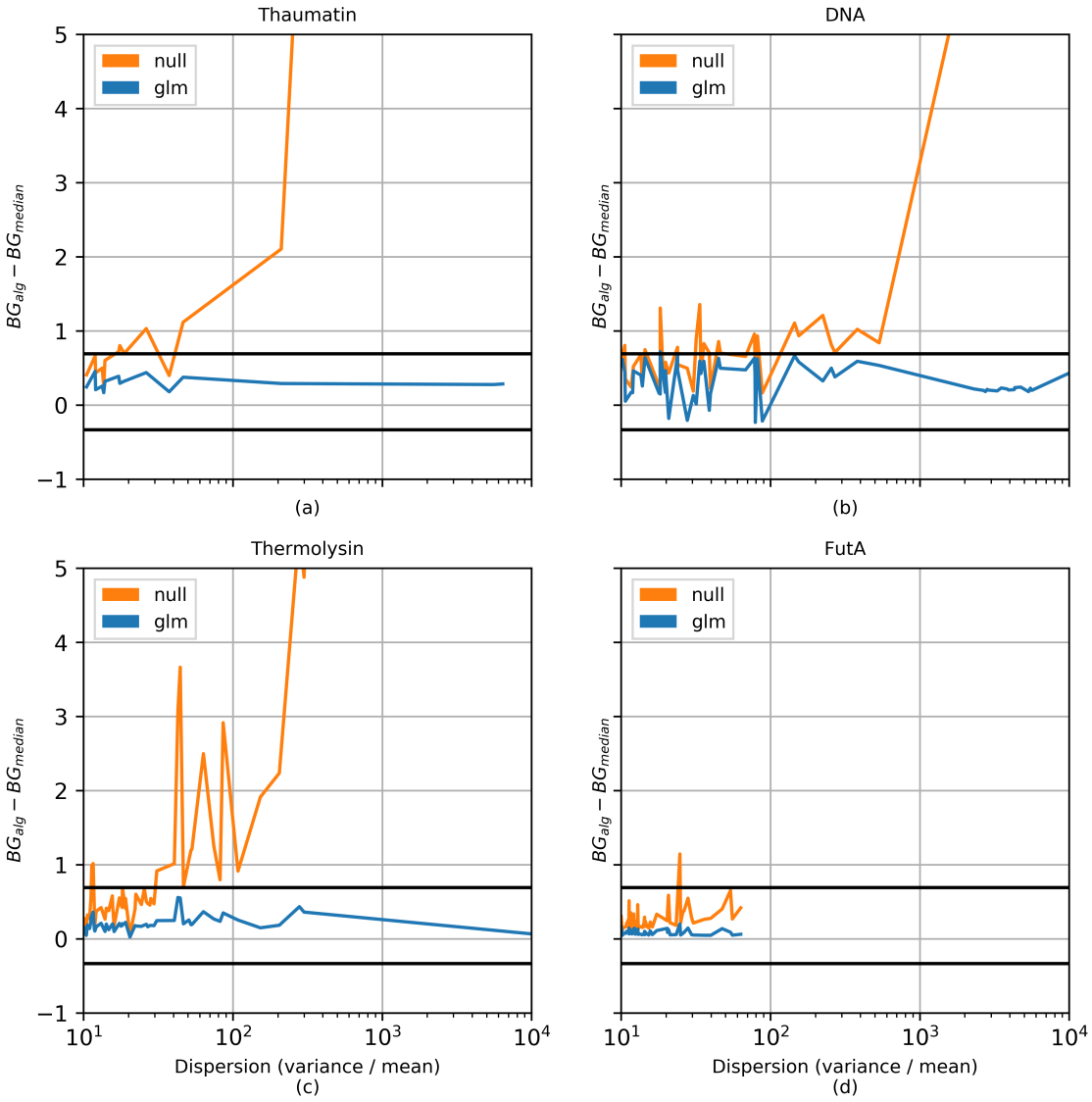


Figure 3.4: The difference between the estimated background value with either no outlier handling or with the GLM algorithm, and the median (*i.e.* most robust) background estimate for Bragg reflections with large indices of dispersion in the surrounding background pixels (an indication of the presence of pixel outliers) for (a) Thaumatin, (b) DNA, (c) Thermolysin and (d) FutA. The horizontal black lines in each plot are at ln(2) and −1/3; for a Poisson distribution, the bounds on the median are $\lambda - \ln(2) <= \text{median} < \lambda + 1/3$ (Choi, 1994).

Table 3.2: The twin fractions deduced from the L-test (column label "L") and the 4^{th} moments test (column label "M") reported by *CTRUNCATE* for each dataset processed using each outlier handling algorithm.

	Thaumatin		DNA		Thermolysin		FutA	
Algorithm	L	M	L	M	L	M	L	M
truncated	0.04	0.00	0.50	0.28	0.50	0.23	0.04	0.00
nsigma	0.50	0.27	0.50	0.50	0.50	0.50	0.50	0.30
tukey	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.32
plane	0.06	0.01	0.50	0.42	0.50	0.50	0.15	0.11
normal	0.50	0.30	0.50	0.50	0.50	0.50	0.39	0.27
glm	0.03	0.00	0.04	0.00	0.03	0.00	0.03	0.00
null	0.03	0.00	0.05	0.00	0.03	0.00	0.03	0.00

3.3.4 Effects on data reduction

Since the background values are systematically underestimated for many of the algorithms, the intensities of the reflections are systematically overestimated. This impacts on the distribution of observed reflection intensities resulting in the appearance of too few weak reflections being recorded. This can cause problems with statistics that test for twinning in the data (Yeates, 1997). Two such statistics are the L-test (Padilla and Yeates, 2003) and the moments test (Stein, 2007). Table 3.2 shows the twin-fractions resulting from application of the two twinning tests as implemented in *CTRUNCATE* for each dataset and for each outlier handling algorithm. Table 3.2 shows that, in most cases, the traditional outlier handling algorithms introduce the appearance of twinning to varying degrees. In contrast, for the data processed with no outlier handling, and for the GLM method, this effect is consistently absent.

The impact on the distribution of intensities is illustrated in more detail by Figures 3.5 and 3.6. Figure 3.5 shows the cumulative distribution function for $|L|$ as produced by *CTRUNCATE* for each dataset and each outlier handling method. For clarity, the results from the GLM algorithm are shown in a separate plot in each case. Figure 3.6 shows the 4^{th} acentric moments of E, the normalised structure factors, against resolution for each dataset processed with each outlier handling method.

For error free data, the 4^{th} acentric moment of the normalised structure factors against resolution takes on a value of 2 for untwinned data and 1.5 for perfectly twinned data (Stein, 2007). When the variances on the intensities are taken into account, the value of the moments is inflated by $\sigma(I)^2/< I >^2$; this is shown by the black theoretical curve in Figure 3.6; this curve was generated by the *Phaser* program (McCoy *et al.*, 2007). Here we can see that as resolution increases, the data based on traditional methods show a reduced spread in the distribution of intensities which

may be interpreted as increasing amounts of twinning. In reality, the plot probably results from a dual effect. The background level decreases at high resolution, so the effect of the bias in the background estimates becomes increasingly pronounced. At the same time, the intensity of the reflections also decreases at high resolution meaning that the relative effect of the systematically lower background estimates are amplified. In contrast, the GLM method shows the expected behaviour. At low resolution, the 4^{th} moment is around 2, indicating no twinning. At high resolution, the moments increase as expected due to the decreasing signal-to-noise ratio; the increase follows the theoretical curve.

3.4 Conclusion

The use of a robust generalised linear model algorithm for the estimation of the background under the reflection peaks in X-ray diffraction data has been presented. Traditional methods for handling pixel outliers systematically underestimate the background level and consequently overestimate the reflection intensities even in the absence of any pixel outliers in the raw data. This can cause statistical tests to give the false impression that a crystal is twinned. The GLM method used here is robust against such effects. When no outliers are present, the estimates given by the GLM algorithm are, on average, the same as those with no outlier handling; the mean normalised difference between the estimates derived from the GLM method and those with no outlier handling are -3.67×10^{-5}, -8.38×10^{-4}, 3.38×10^{-4} and -2.93×10^{-3} for the Thaumatin, DNA, Thermolysin and FutA datasets respectively. When outliers are present, the method gives values within the expected bounds of the median. The method is implemented in *DIALS* and is currently the default algorithm when run stand-alone or through *xia2*. This work was published in Parkhurst *et al.* (2016).

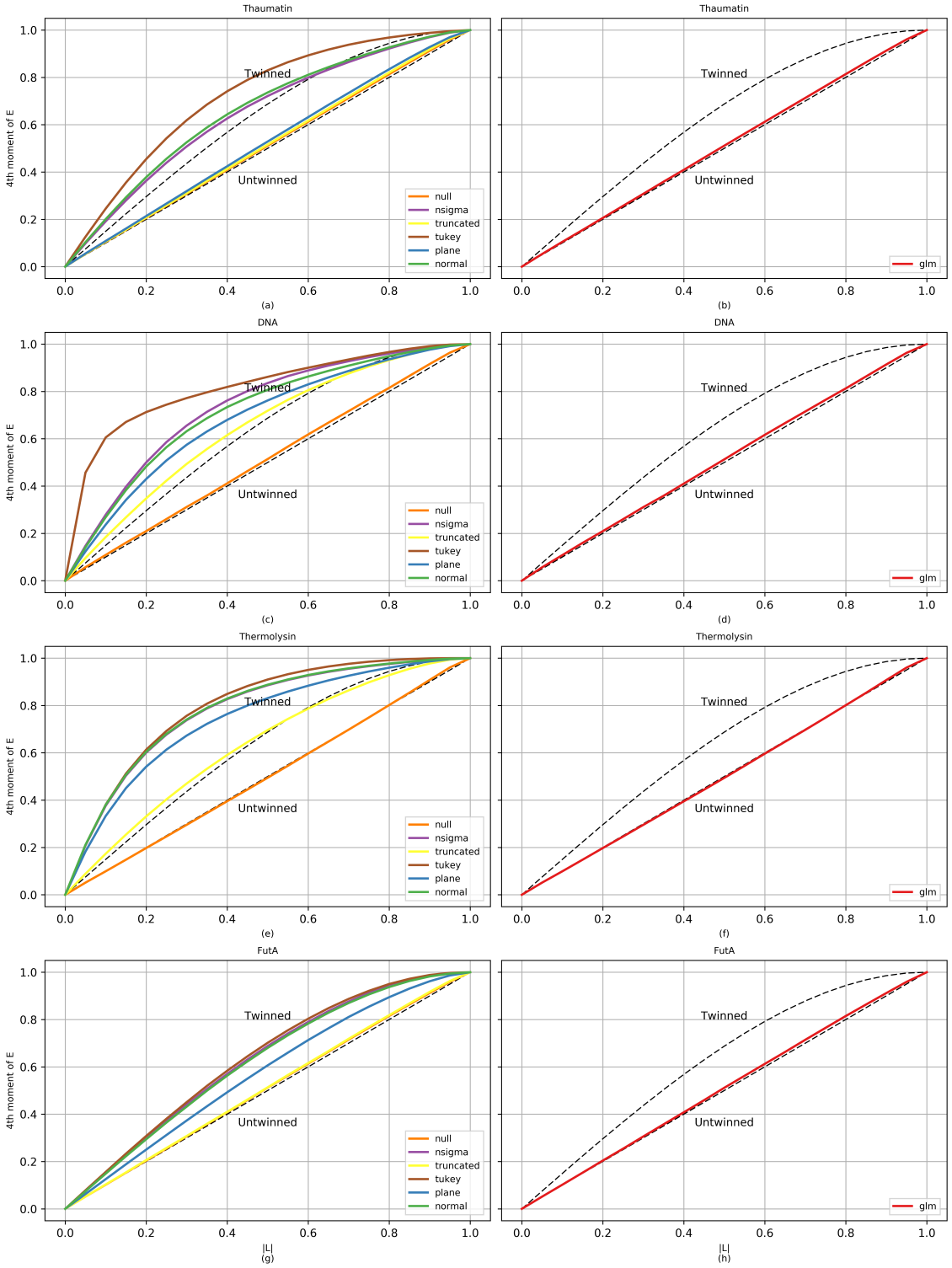


Figure 3.5: Cumulative distribution function for $|L|$ for Thaumatin with (a) the traditional outlier handling methods and (b) with the GLM method, for DNA with (c) the traditional outlier handling methods and (d) with the GLM method, for Thermolysin with (e) the traditional outlier handling methods and (f) with the GLM method, and for FutA with (g) the traditional outlier handling methods and (h) with the GLM method.

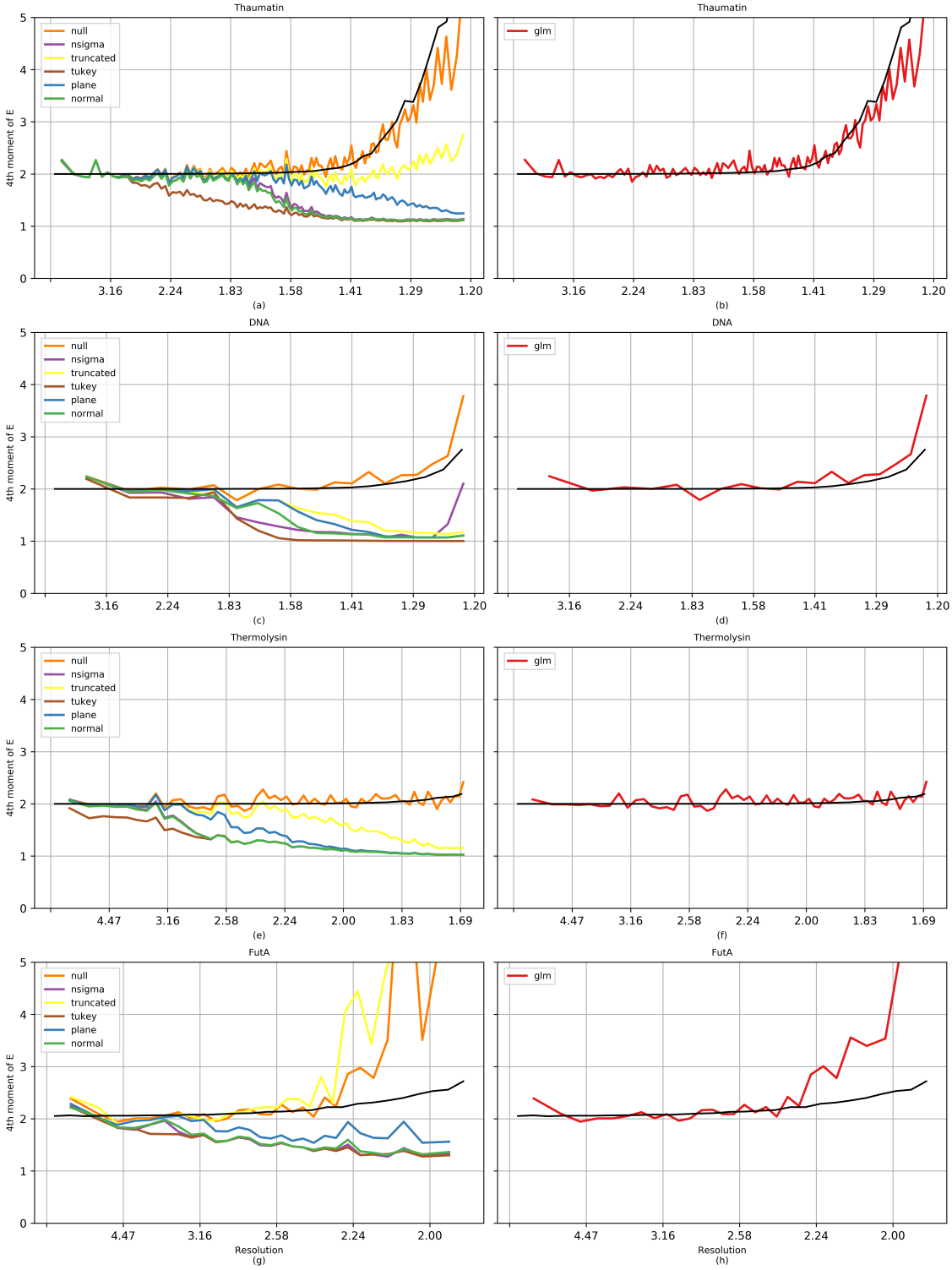


Figure 3.6: 4th acentric moment of E vs resolution for Thaumatin with (a) the traditional outlier handling methods and (b) with the GLM method, for DNA with (c) the traditional outlier handling methods and (d) with the GLM method, for Thermolysin with the (e) traditional outlier handling methods and (f) the GLM method, and for FutA with the (g) traditional outlier handling methods and (h) the GLM method. The theoretical curve for the acentric moments is shown in black.

Chapter 4

Background modelling in the presence of ice-rings

4.1 Introduction

In macromolecular crystallography (MX), for data collected using the rotation method, a dataset is typically composed of a sequence of X-ray diffraction images (Arndt and Wonacott, 1977); each image covers a fixed oscillation and, as the crystal is rotated, individual reflections enter and subsequently exit the diffracting condition. Integration programs - such as *MOSFLM* (Leslie, 1999), *XDS* (Kabsch, 2010b), *d*TREK* (Pflugrath, 1999), *HKL2000/DENZO* (Otwinowski and Minor, 1997) and *DIALS* (Winter *et al.*, 2018) - are used to predict where each Bragg reflection will appear on the detector and then to provide an estimate of each reflection's intensity. The simplest method for computing the reflection intensities is *via summation integration*; most integration programs provide an implementation and whilst details may differ, the procedure is generally the same.

1. First, the location and extent of each reflection on the detector is predicted and pixels are assigned as either *foreground* or *background* depending on whether they are predicted to contain signal from the Bragg reflection or not.

2. The background under the reflection peak is then estimated from the surrounding *background* pixels since it is not possible to measure the background under the peak directly. As such, a model of background is required and the model fitted to the *background* pixel data.

3. Finally, the reflection intensity is estimated by summing the total counts in the *foreground* region and subtracting the sum of the estimated background counts.

In most integration programs, simple background models have been employed; a major reason for this is the necessity of having a computationally efficient implementation since the background needs to be estimated for a large number of reflections in each dataset. The best way to model the general reflection background is not always obvious since the background varies considerably between datasets. As such, the background under each reflection peak is often assumed to be a constant value (Kabsch, 2010a) or a plane with a small gradient (Rossmann *et al.*, 1979; Otwinowski and Minor, 1997; Leslie, 1999). In *DIALS*, either a constant or planar background can be used (Parkhurst *et al.*, 2016). These simple models have been employed with great success for many years (Diamond, 1969; Otwinowski and Minor, 1997; Leslie, 1999; Kabsch, 2010a) since, in a typical MX X-ray diffraction dataset, individual reflections extend over a small number of pixels and the local X-ray background is usually relatively flat.

Whilst such a simple background model may be appropriate in the majority of cases, particularly for well-measured data, it is not applicable where the background changes significantly over the extent of a single reflection peak. In such cases a flat or planar background model is likely to provide an inaccurate estimate of the background in the reflection peak region. Large variation in the background counts can be the result of various effects such as scattering from the cryostream nozzle or, in serial crystallography, from the linear jet that transports crystals into the beam which creates a streak of diffraction perpendicular to the jet direction. Large variations are also often seen around the backstop; however, these reflections are usually omitted from processing due to their large Lorentz factor. Perhaps the most common pathology seen in diffraction images resulting in a large variation in background counts is the presence of water ice rings (Mitchell and Garman, 1994). A detailed description of the theoretical manifestation of cubic and hexagonal ice (the most common forms) in diffraction images can be found in Thorn *et al.* (2017). In practice, when cubic ice diffraction is observed, hexagonal ice diffraction is also observed (Fuentes-Landete *et al.*, 2015).

If the background is assumed to be locally flat, but ice rings are present, the reflection intensities will be systematically biased. The effect on the background estimation caused by the presence of ice rings can be readily seen by plotting the scaled reflection intensities as a function of resolution, as shown in Figure 4.1. This plot shows a large spike in the reflection intensities at ice ring resolutions, with a drop in the reflection intensities either side of the ice ring; *i.e.* the presence of ice rings causes both systematic over- and under-estimation of the reflection intensities at characteristic resolutions. Indeed, at high-resolution, where the true reflection intensities are very small, the positive systematic bias in the background estimate causes the average reflection intensity to be less than zero at resolutions immediately

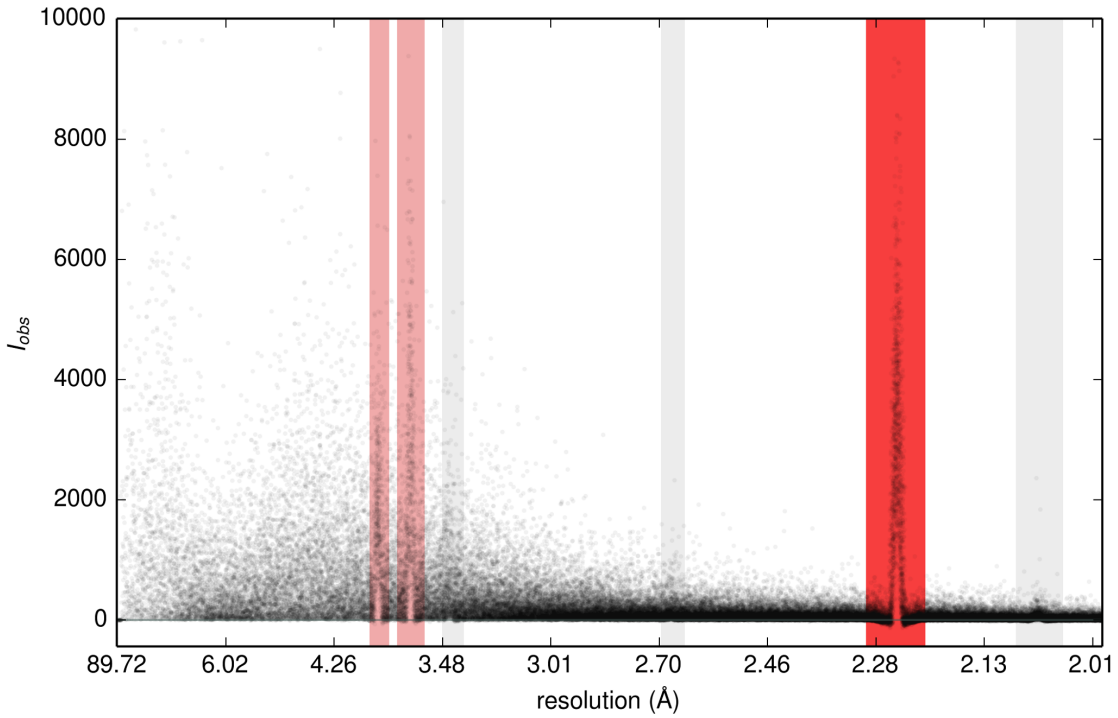


Figure 4.1: Intensity vs resolution for a dataset with strong ice rings. Such plots can be readily generated by *AUSPEX* (Thorn *et al.*, 2017). The points show the intensities for individual reflections. The characteristic ice ring resolutions are shown here in grey with automatically detected ice rings being flagged in red. Spikes in reflection intensity are observed at ice ring resolutions indicating bias in the background determination.

either side of the ice rings.

This can be explained by considering the application of a simple background model to a reflection positioned close to an ice ring as illustrated in Figure 4.2. As the ice ring intrudes into the background region of the reflection shoebox, the background level in the reflection peak region is over-estimated due to the higher valued counts from the ice ring. When the reflection foreground covers the peak of the ice ring, then the background region of the reflection shoebox contains pixels with fewer counts than should be modelled in the reflection peak. Consequently, the background in the reflection peak will be under-estimated and the reflection intensity will be over-estimated. This leads to many reflections being rejected as outliers during data reduction and thus a loss of information. This effect is more pronounced for sharper ice rings; however, on average, fewer reflections will be affected than in the case of more diffuse ice rings which cover a larger resolution range.

For most integration programs, the handling of ice rings and other complex background features is problematic and proper modelling is rarely attempted. Some programs, such as *MOSFLM* and *XDS*, provide parameters to exclude reflections within a user-specified resolution range (*i.e.* those falling on ice rings). Therefore, in these programs, reflections falling on ice rings can be easily excluded from the

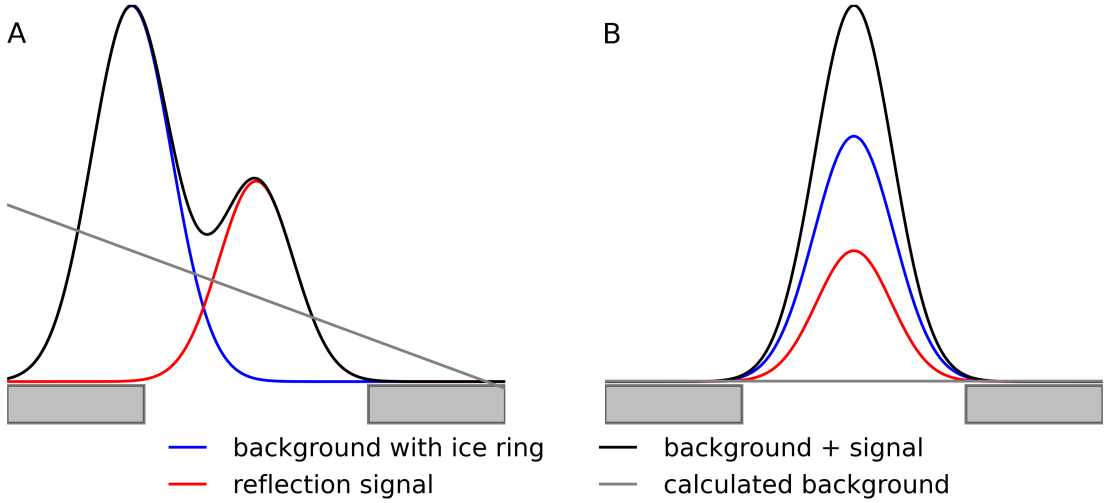


Figure 4.2: Illustration of the effect of ice rings on the background determination when a simple plane model is employed. The shaded rectangles indicate the background pixels used to estimate the background. When the reflection is centred on the tail of the ice ring (A) the background is over-estimated. When the reflection is centred on the peak of the ice ring (B) the background is under-estimated.

processing if desired; however, this usually results in a loss of otherwise potentially useful information. In *d*TREK* and *HKL2000/DENZO* (Otwinowski and Minor, 1997) parameters are provided to remove reflections whose background counts vary excessively; however, this will also result in information loss. It is often the case that the reflections recorded on ice rings are handled during scaling rather than integration. This is particularly the case at higher-resolution where ice rings may not be immediately visible on single detector images. Scaling programs such as *AIMLESS* (Evans and Murshudov, 2013) have outlier handling routines that exclude intensity measurements that are not consistent between symmetry equivalent reflections. Additionally, a resolution range can be set to exclude reflections from the scaling. Programs such as *CTRUNCATE* (Winn *et al.*, 2011), *phenix.xtriage* (Zwart *et al.*, 2005) and *AUSPEX* (Thorn *et al.*, 2017) can be used to automatically determine - from the scaled reflection data - whether the data have been contaminated by ice rings.

An attempt to handle ice rings external to the integration program is described by Chapman and Somasundaram (2010). They describe a method to subtract the ice ring intensity from the raw image data as a pre-processing step before integration. However, this approach is not ideal since the statistics of the data will be altered. Furthermore, the shape of the ice rings are assumed to be radially Gaussian with resolution and perfectly circular which may not be the case in practice. The data as recorded by a photon-counting detector is "count data" which is well modelled by a Poisson distribution. The Poisson distribution is discrete and only valid for

positive pixel counts. Subtracting the background prior to the integration will result in the data no longer being Poisson distributed with some pixels possibly containing negative counts and others containing a non-integer number of counts. This will invalidate assumptions about the statistical properties of the data in the integration program and impact the estimation of the errors in the intensities. For this reason the ice ring background should be modelled explicitly during the reflection integration step.

4.2 Algorithm

In this chapter, a new algorithm for modelling the X-ray diffraction background in the presence of ice rings is described. The algorithm consists of two distinct steps; first a global model of the background at each image pixel is generated, then the model is fit locally and independently for each predicted reflection in the dataset. The algorithm is implemented within the *DIALS* framework and the program usage is given in Appendix A.2.

4.2.1 Global background model

The current implementation, within *DIALS* (Winter *et al.*, 2018), is restricted to a static model that is applied to reflections over the entire rotation scan. For the static background model algorithm, let us assume that the shape of the background model remains fairly stable across all the images in the dataset. In the case where the background is contaminated with ice rings, an approximate model should perform better than a flat background model; this therefore represents an improvement in the handling of data with complex background.

The global background model is calculated as the mean value at each pixel, averaged over all images in the dataset. This method for generating the global background model is computationally efficient and simple to compute; care needs to be taken to ensure that the inclusion of outlier pixels does not cause the background model to be distorted. In this context, outlier pixels are considered to be pixels which contain intensity from predicted reflections as well as unmodelled intensity which may come from reflections whose extent is badly predicted, zingers (random spikes in intensity from, for example, cosmic rays), or other sources.

Intensity from predicted reflections is handled by generating a mask for each image delineating the foreground and background for each reflection, and then to only use background pixels for the global background model. The mask contains *True* where the pixels are predicted to only contain background counts and *False* where they are predicted to contain intensity from predicted reflections. Once the

process concludes and a mean value is computed at each pixel, the number of images contributing to the mean for each pixel is calculated. A second pixel mask is then generated containing *True* where the number of contributed images is greater than some user-specified value (by default 10) and *False* otherwise. In this way, pixels where only a small number of images have contributed are excluded. Where the number of images in the dataset is less than 10, the minimum number of required images is reduced; however, the method is most effective where a larger number of images is available.

In order to ensure that the model is not affected by outliers caused by unmodelled intensity, a number of filters are applied to the mean image to produce the final background model as follows:

1. Firstly, the mean image and mask are transformed into a polar image such that columns in the transformed image correspond to lines of constant resolution. This requires that each pixel in the raw untransformed image is mapped onto the transformed grid. In the implementation described here, this is done by computing the overlap of each pixel in the raw image with the transformed grid and then using a polygon clipping algorithm (Sutherland and Hodgman, 1974) to compute the overlapping area between the pixel and the grid. This fractional overlap is used to determine the fraction of counts in each pixel that is distributed to each grid point in the transformed image. The number of counts in the raw and transformed image is then conserved. The benefit of applying filtering to this transformed image rather than the raw mean image is that, in the case of ice rings in particular, the background is likely to vary less along lines of constant resolution. Therefore, the variation along columns in the transformed image is likely to be small and most variation will occur along the rows which correspond to increasing resolution. The polar transform will tend to sample pixels at high resolution less finely than at low resolution resulting in smoothing along lines of constant resolution.

2. Median filtering is then applied to the columns of the transformed image such that for each pixel, the median of the neighbouring N (by default 10) pixels along each column (wrapped at the column ends) is used. This has the effect of removing the effect of unwanted outliers, in particular high-valued pixel outliers.

3. The transformed image will contain pixels that are masked out; these need to be filled in order to provide full coverage of the detector for the background model. Again, since the image has been transformed and the variation along columns is small, the missing pixel values can be filled using a simple iterative diffusion

algorithm based on application of Laplace's equation with Dirichlet boundary conditions whereby missing pixels are iteratively filled with the values derived from adjacent pixels until convergence is achieved (Guillemot and Le Meur, 2014). The missing pixels could also be filled by fitting, for example, a 2D spline surface.

4. Finally, the polar image is transformed back *via* the same polygon clipping process with the counts in the transformed image being redistributed to the image in the original coordinate system. Application of this gridding procedure will result in some additional smoothing in the processed image. The final result is a smoothly varying global background model.

4.2.2 Maximum likelihood fitting for each reflection

The background is fit to each reflection locally and independently by simply scaling the background model to fit the counts in the background region of the reflection in question. The pixel counts are assumed to be drawn from a Poisson distribution. The terms used in the following equations are given in the list of symbols on page 13. For each pixel, i, in the reflection background, consisting of N pixels, the probability of observing c_i counts, given the background model, b_i, scaled by the parameter B is as follows:

$$P(c_i|B, b_i) = \frac{(Bb_i)^{c_i} \exp(Bb_i)}{c_i!}. \tag{4.1}$$

The value of the parameter, B, is then estimated *via* maximum likelihood by considering the joint probability distribution over the N pixels.

$$L = \prod_{i=1}^{N} \frac{(Bb_i)^{c_i} \exp(Bb_i)}{c_i!}. \tag{4.2}$$

Using the log likelihood and taking derivatives with respect to the scale parameter, $\partial \log(L)/\partial B = 0$, results in a very simple and computationally efficient equation for the scale factor, B, for each reflection which is simply:

$$B = \frac{\sum_{i=1}^{N} c_i}{\sum_{i=1}^{N} b_i}. \tag{4.3}$$

However, this equation for the scale factor is not resistant to pixel outliers in the reflection background which must be handled to ensure that the background estimates are reliable. Since the data are Poisson distributed, a principled approach to the modelling of the background in the presence of pixel outliers would be to use a robust generalised linear model (GLM) algorithm (Parkhurst *et al.*, 2016); however,

whereas the robust GLM algorithm can be made computationally efficient for the case of a flat background model, it is difficult to optimise for more complex models. Since the background needs to be estimated for a large number of reflections in each dataset, computational efficiency is a requirement of any background modelling algorithm during integration; therefore, a simpler approach was taken in this case. The Anscombe variance stabilising transform for a Poisson distribution (Anscombe, 1948), given by $y = 2\sqrt{x + 3/8}$, was used to transform the Poisson distributed data such that it is approximately normally distributed with a variance of 1. This transformation is biased where the Poisson scale parameter is very small (< 4); however, in the case of data where the background is contaminated with ice rings, the background is generally much larger and this approximation may be used. The robust estimation is then performed using the Huber weighting function (Huber, 1964) such that for a pixel, i, with transformed value y_i, predicted value μ_i, variance $v_i = 1$ and residual $r_i = (y_i - \mu_i)/\sqrt{v_i}$, the pixel weighting, w_i, will be given by:

$$w_i = \begin{cases} 1, & |r_i| \leq c \\ c/|r_i| & |r_i| > c \end{cases}. \tag{4.4}$$

This weighting function has the effect of damping values outside a range defined by the tuning constant, c, whose default value is 3 (*i.e.* transformed pixel values greater than 3 standard deviations from the mean are damped). The quasi-likelihood equation implementing this robust algorithm is then solved using iteratively reweighted least squares.

4.2.3 Robust M-estimator for background scale factor

In robust estimation, M-estimators minimise a function of residuals of the following form:

$$min \sum_{i=1}^{n} \rho(r_i). \tag{4.5}$$

The value of each residual, r_i, depends at each iteration on the value of the parameter estimates, $\boldsymbol{\beta}$. Taking derivatives with respect to each parameter, β_j, gives:

$$\sum_{i=1}^{n} \frac{\partial \rho(r_i)}{\partial r_i} \frac{\partial r_i}{\partial \beta_j} = 0. \tag{4.6}$$

Weights are then defined as:

$$w_i = \frac{1}{r_i} \frac{\partial \rho(r_i)}{\partial r_i}. \tag{4.7}$$

The equation to be solved then becomes:

$$\sum_{i=1}^{n} w_i r_i \frac{\partial r_i}{\partial \beta_j} = 0. \tag{4.8}$$

The parameter estimates can be found at each iteration as:

$$\boldsymbol{\beta}^{(t+1)} = (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W} \boldsymbol{y}. \tag{4.9}$$

In order to fit the global background model, a single scale factor is used. The design matrix, $\boldsymbol{X}$ then has a single column. Therefore, each iteration can then be simplified to the following equation:

$$\beta^{(t+1)} = \frac{\sum_{i=1}^{n} x_i w_i y_i}{\sum_{i=1}^{n} x_i w_i x_i}. \tag{4.10}$$

In our implementation, the Huber function is used which gives the robust function of residuals as follows:

$$\rho(r_i) = \begin{cases} {r_i}^2/2, & |r_i| \leq c \\ c(|r_i| - c/2) & |r_i| > c \end{cases}. \tag{4.11}$$

This results in the following weighting function:

$$w_i = \begin{cases} 1, & |r_i| \leq c \\ c/|r_i| & |r_i| > c \end{cases}. \tag{4.12}$$

4.3 Analysis

4.3.1 Experimental data

In order to evaluate the effect on the quality of processed data when there are prominent ice rings in the X-ray background, some datasets were selected from the Joint Centre for Structural Genomics (JCSG) (Gabanyi *et al.*, 2011). Whilst, the method also applicable to data collected on other detectors, only datasets collected using a DECTRIS PILATUS detector (Henrich *et al.*, 2009) were considered for analysis. Datasets were chosen manually by inspecting a plot of the intensity *versus* resolution using *AUSPEX* (Thorn *et al.*, 2017); those datasets showing a noticeable systematic bias at ice ring resolutions were used (see Figure 4.6). Two datasets (4MJG and 4PUC) were identified for which reflections in entire resolution ranges corresponding to ice rings had been discarded in deposition; in the following

analysis, the data were processed without omission. 13 datasets that showed ice ring pathologies and which were successfully processed using *DIALS* inside the *xia2* (Winter, 2009) automatic processing pipeline were used in the analysis. Table 4.1 shows the datasets used in more detail, giving the known space group and resolution. Further details about the data processing, data reduction and refinement are given in Appendix A.3.

4.3.2 Refinement results

The R_{work} and R_{free} statistics reported by *REFMAC5* for each dataset as processed with both the default background algorithm and global background algorithm are shown in Table 4.1; additionally, R_{free} for each dataset is shown in Figure 4.3. The improvement in R_{work} and R_{free} is shown for both summation integrated data and profile fitted data. It can be seen that in each case, both R_{work} and R_{free} are reduced by the use of the global background model algorithm over the default background algorithm. An improvement is seen when the data is processed using both summation integration and profile fitting. In some cases (*e.g.* 4KW2 and 4OPM) the improvement is minor; however, in others, such as 4PUC, the improvement in the refinement R factors is dramatic, with R_{free} being reduced by 4.9%. In the case of 4PUC, as previously reported, reflections from entire resolution ranges around ice rings were omitted in the deposited data (the completeness of the deposited data was 78.1%, the completeness of the data processed here is 99.3%). In general, most datasets see a moderate improvement in the R_{free}; the mean improvement in R_{free} across all datasets was 2.0% when using summation integration and 1.1% when using profile fitting. Profile fitting also consistently results in lower R_{work} and R_{free} than summation integration and the improvement in R_{work} and R_{free} when using the new global background model algorithm is slightly lower than the improvement seen with data processed using summation integration.

4.3.3 Case studies

Of the 13 JCSG datasets processed above, three were selected for more detailed analysis. The first image and the average background *versus* resolution for each of the datasets is shown in Figure 4.4. The datasets were chosen as follows:

- **4PUC**. This dataset shows an example of very strong and prominent ice rings. The ice rings in this dataset are narrow and the three inner rings corresponding to hexagonal ice rings can be clearly distinguished in the diffraction images. Handling reflections falling on these ice rings is likely to be a challenge for current background modelling algorithms. Each image in the dataset covered

Table 4.1: A list of JCSG datasets with ice ring pathologies. The improvement in R_{free} reported by *REFMAC5* for data integrated using the global background model algorithm over data integrated with the default background algorithm is given for profile fitted intensities (prf) and summation intensities (sum). For each dataset, the same set of integrated reflections were used for the different sets of processing. Here $\Delta = R_{default} - R_{global}$; a positive value indicates an improvement using the global background model algorithm. For brevity, columns with data from the default background algorithm are labelled "D" and columns with data from the global background algorithm are labelled "G". The completeness is shown for the data processed (*DIALS*) and the completeness reported in the PDB. For 4PUC, an R_{free} of 19.98% is reported in the PDB. For this dataset, reflections falling on ice rings were excluded from processing resulting in low completeness. Refining the subset of reflections present in the deposited data processed using the new background algorithm against the deposited structure resulted in an R_{free} of 19.46%.

				Completeness (%)		$R_{free}(sum)$ (%)			$R_{free}(prf)$ (%)		
PDB ID	SG	Resol. (Å)	Multipl.	*DIALS*	PDB	D	G	Δ	D	G	Δ
4DN6	$P4_22_12$	2.80	12.6	100.0	99.0	35.1	34.0	1.1	34.2	33.1	1.1
4E6E	$P3_221$	2.12	12.5	99.9	99.6	29.7	26.8	2.8	26.7	25.6	1.1
4EF1	$P12_11$	1.90	3.4	98.0	97.8	34.7	31.9	2.8	31.7	30.2	1.5
4EPZ	$C222_1$	1.68	4.2	98.5	98.0	25.1	22.6	2.5	22.2	21.4	0.8
4EZG	$P2_12_12_1$	1.50	5.3	98.8	99.0	23.5	22.2	1.3	20.2	19.8	0.4
4FMR	$P12_11$	2.25	6.9	97.2	97.9	26.4	25.0	1.4	25.5	24.6	1.0
4HF7	$C222_1$	1.77	9.8	99.7	98.2	36.9	32.1	4.8	28.9	27.5	1.4
4IEJ	$P6_122$	1.45	8.3	100.0	99.8	30.9	29.5	1.3	27.2	26.6	0.6
4KW2	F432	2.32	110.4	100.0	99.8	23.9	23.8	0.1	23.0	22.9	0.1
4MJG	$P3_221$	2.65	8.2	99.9	88.7	29.7	28.3	1.4	28.3	27.6	0.7
4OPM	C121	1.70	7.0	99.6	97.1	23.4	22.7	0.7	20.9	20.7	0.2
4PS6	$P12_11$	1.25	6.2	93.7	85.8	21.5	20.3	1.2	18.4	17.9	0.5
4PUC	$P2_12_12_1$	2.00	9.2	99.3	78.1	29.8	25.4	4.4	28.1	23.2	4.9

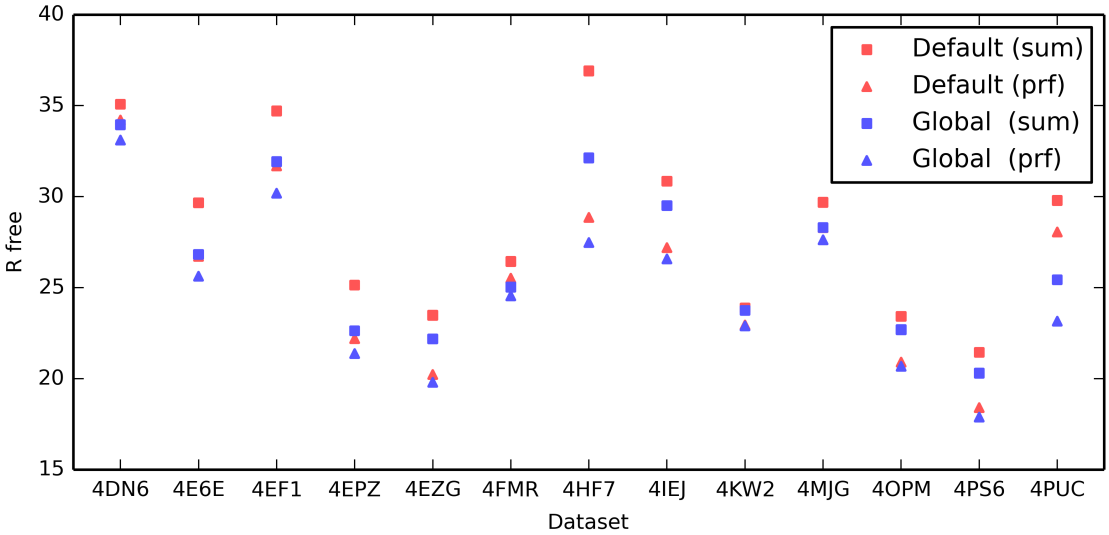


Figure 4.3: The R_{free} after refinement using the *REFMAC5* program for both summation integration (sum) and profile fitting integration (prf) using both the default background algorithm assuming a flat background model and the global background model algorithm. In all cases, the global background model results in an improvement in the R_{free} factors from refinement. Using profile fitting (prf) instead of summation (sum) also results in an improvement in R_{free} in each case.

a rotation of 0.25 degrees. This is a MAD dataset consisting of 3 sweeps at different wavelengths. In the data processing, all 3 sweeps were used and merged together.

- **4EF1**. This dataset shows a moderate improvement in the R factors. The dataset has ice rings from nano crystalline cubic ice. Each image in the dataset covered a rotation of 0.3 degrees.

- **4KW2**. This dataset shows the smallest improvement in the R factors. The dataset has ice rings from nano crystalline cubic ice. Each image in the dataset covered a rotation of 0.5 degrees. This is a MAD dataset consisting of 18 sweeps at different wavelengths. In the data processing, all 18 sweeps were used and merged together.

4.3.4 Pixel statistics

During the creation of the global background model, the mean, variance and index of dispersion (variance / mean) are calculated independently for each pixel across all images in the rotation scan. Note that pixels predicted to contain intensity from reflections are not used in the calculation of these images. The mean and index of dispersion are shown for each dataset in Figure 4.5. From a qualitative inspection, the mean background image visually resembles a smoothed version of the raw image data shown in Figure 4.4. The dispersion images, however, indicate that the variation in the background is not uniform across the detector surface. In particular, background pixels not containing ice rings appear to vary very little across the rotation scan, as indicated by the index of dispersion being close to 1.0. By contrast, pixels containing ice rings appear to show much greater variation across the scan, with an index of dispersion of greater than 2.0 in some cases. This appears to indicate that the intensity of the ice ring background varies much more than the general background counts.

4.3.5 Intensity *versus* resolution

Figure 4.6 shows the intensity plotted against resolution for all reflections in each dataset with the default background algorithm and the new global background model algorithm. In each case, for the default background algorithm, it can be seen that the reflection intensities at ice ring resolutions suffer from systematic bias. This is shown as spikes in the intensity at ice ring resolutions. These spikes are due to the background of reflections lying at ice ring resolutions being under-estimated. So the reflection intensity is over-estimated. Small dips in intensity can be seen either

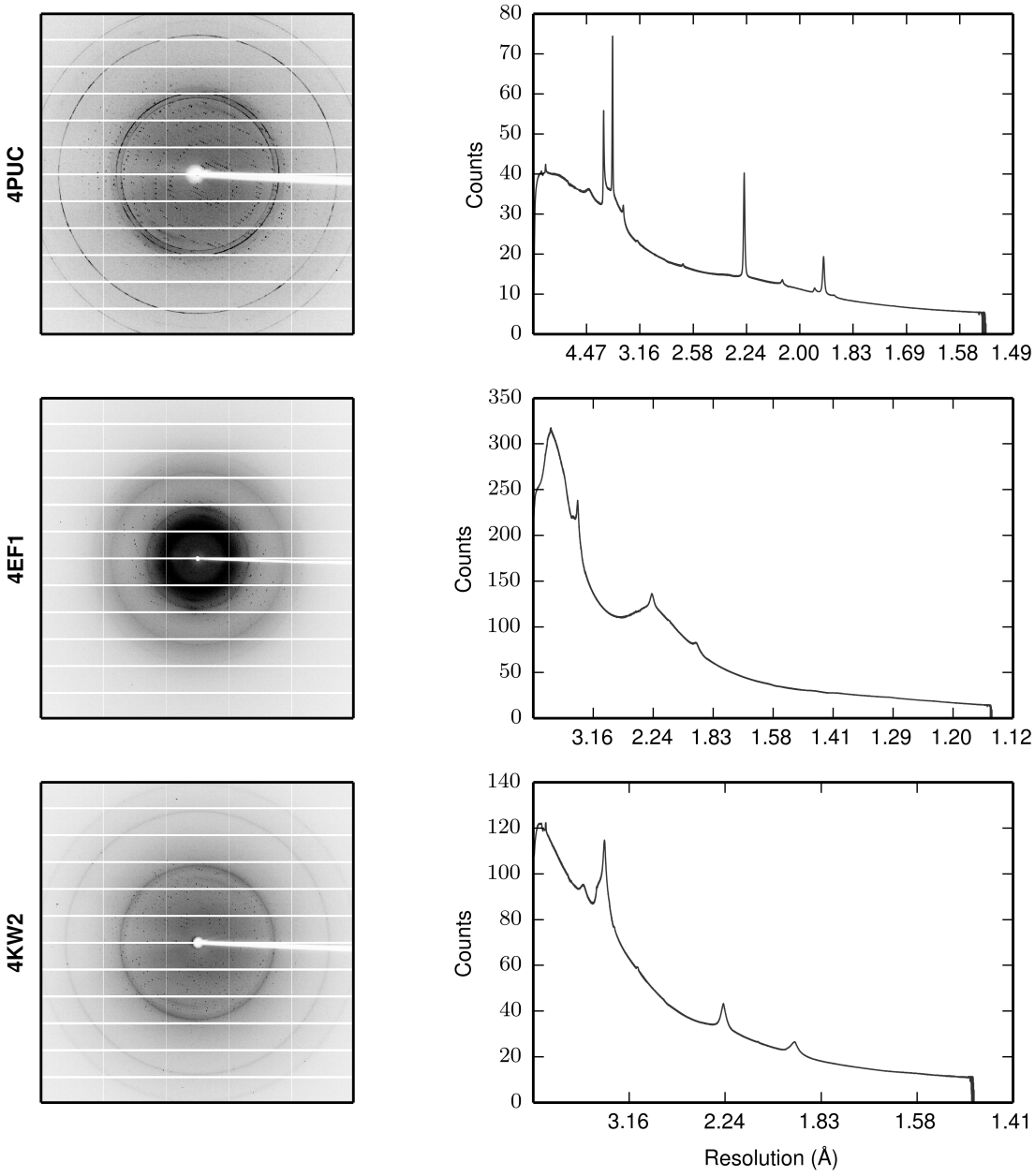


Figure 4.4: The first image for each dataset and the corresponding average background *versus* resolution for 4PUC (top), 4EF1 (middle) and 4KW2 (bottom). Ice rings are visible as discrete or diffuse rings. Note the irregular ice rings in the diffraction image of 4PUC resulting from the dominating ice crystal orientation. Also note that the maximum resolution on each image differs; therefore the ice rings are not always in the same place on the detector.

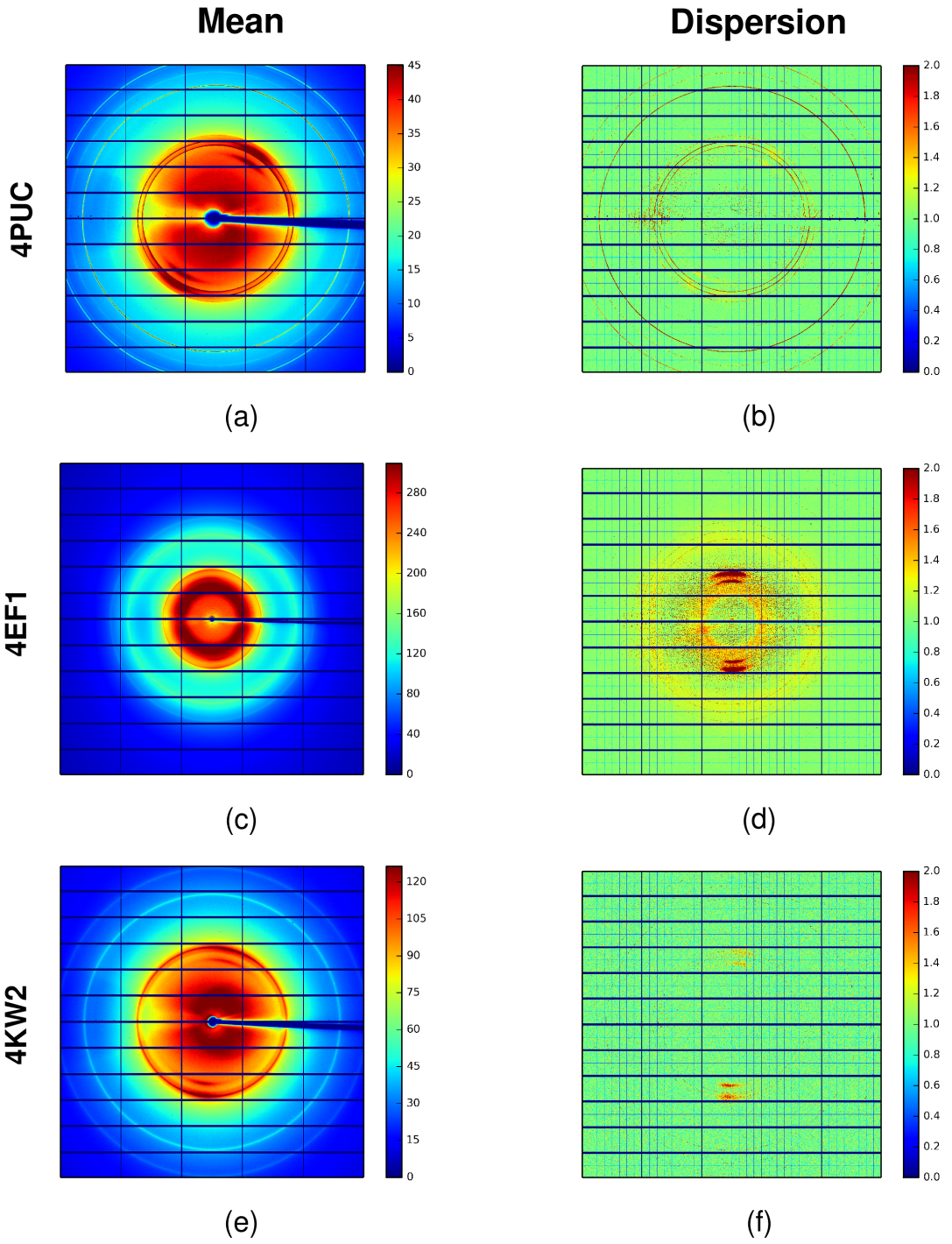


Figure 4.5: The mean and dispersion images for 4PUC (a-b), 4EF1 (c-d) and 4KW2 (e-f). The mean image is the mean value at each pixel through the image stack; the index of dispersion image shows the variation across the dataset at each pixel. In the mean image, the ice rings are clearly visible. The dispersion images also show the structure of the detector. The boundaries between the detector chips are visible as lines of pixels that are under-dispersed relative to a Poisson distribution. This is due to the use of virtual pixels between chips that share counts and whose values are therefore correlated.

side of the ice rings, showing how the background is over-estimated as the ice ring intrudes into the background region of the reflection shoeboxes, thereby causing the reflection intensities to be under-estimated. This is particularly noticeable for dataset 4PUC where the shift is dramatic. Dataset 4EF1 shows a moderate increase in reflection intensities at the ice ring resolutions; 4KW2 only shows a fairly minor shift visible in the ice ring at 3.7Å.

For the new global background model algorithm, the intensity estimates appear to be greatly improved. For the 4EF1 and 4KW2 datasets, the spikes at ice ring resolutions are completely absent, indicating that the systematic bias in the intensity estimates has been reduced relative to the bias for the default background algorithm. For the 4PUC dataset - the most challenging data - there is some improvement; however, peaks at some ice ring resolutions are still present. This is due to the ice ring background being sharp and irregular with time dependent variation throughout the dataset. Taken together, these conditions provide a difficult modelling challenge. The algorithm computes the global background model over a number of images; therefore, the algorithm will tend to perform worse where there are large time dependent variations in the background shape. Nevertheless, 4PUC, showed the best improvement in refinement R-factors as shown in Table 4.1.

4.3.6 Moments of E and R_{free} *versus* resolution

The left panel of Figure 4.7 shows the 4th acentric moments of E, the normalised structure factors, calculated by *CTRUNCATE* (Winn *et al.*, 2011), for each dataset processed with both the default background algorithm and the new global background model algorithm. For error-free data, the 4th moment takes on a value of 2 for untwinned data and 1.5 for perfectly twinned data (Stein, 2007). When the variances on the intensities are taken into account, the value of the moments is inflated by $\sigma(I)^2/< I >^2$ as described in Section 4.3.7; this is shown by the theoretical curve in Figure 4.7; this curve was generated by the *Phaser* program (McCoy *et al.*, 2007). The right panel in 4.7 shows the R_{free} *versus* resolution as reported by *REFMAC5* (Murshudov *et al.*, 2011).

The moment plots seem to mirror those seen in the intensity *versus* resolution plots. Dataset 4PUC shows large deviations from the expected value of 2 at ice ring resolutions with the default background algorithm. After application of the new global background model algorithm the moments, whilst better behaved, still show the effect of the ice rings. For 4EF1, the moments differ at ice ring resolutions for data processed with the default background algorithm and data processed with the new global background algorithm. However, the variation is small relative to the noise. For 4KW2, which showed little improvement after application of the global

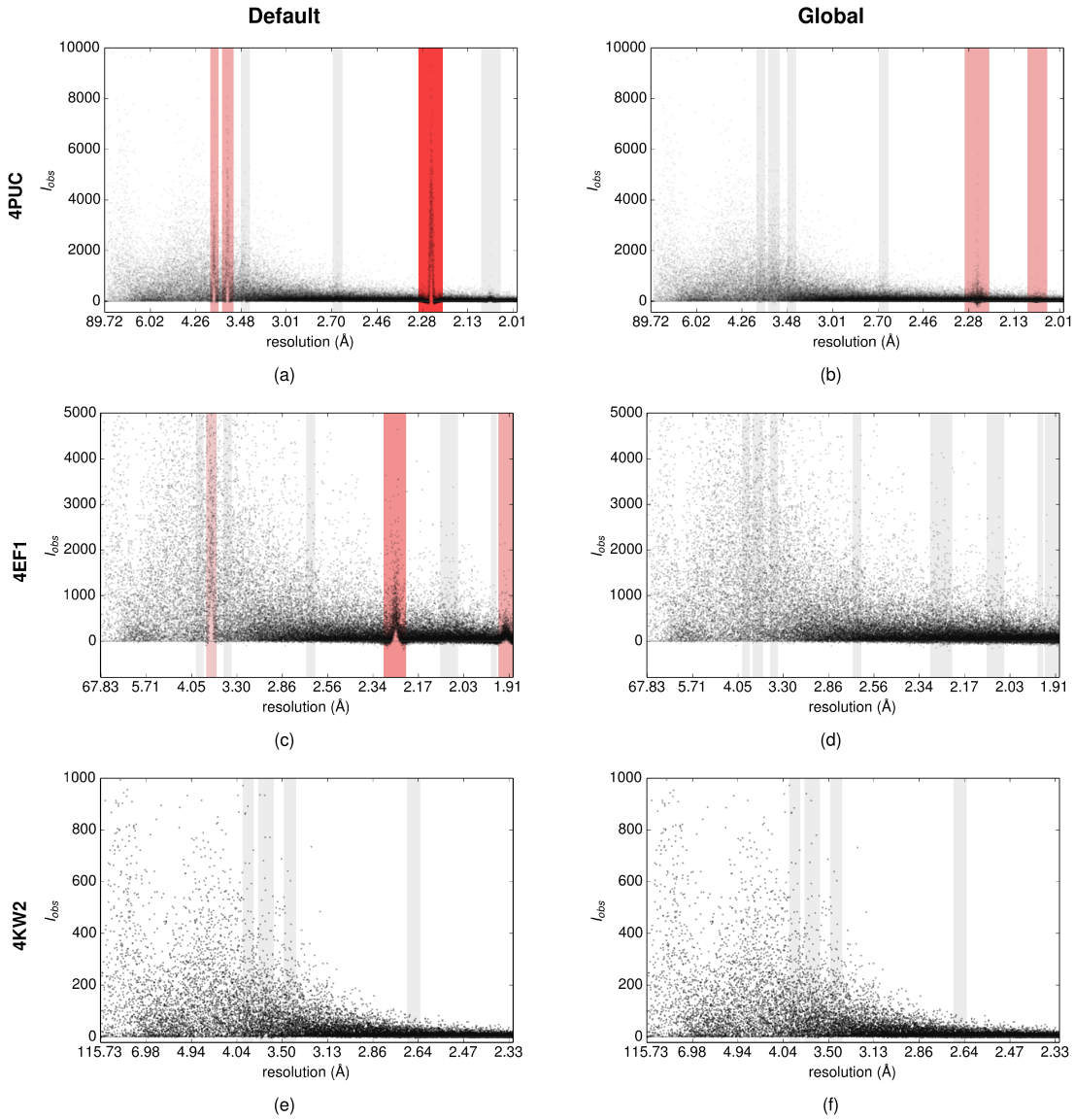


Figure 4.6: The intensity *versus* resolution of reflections processed using the default background algorithm (on the left) and the new global background model algorithm (on the right) for datasets, 4PUC (a-b), 4EF1 (c-d) and 4KW2 (e-f). All plots were generated by *AUSPEX* (Thorn *et al.*, 2017). The points represent the individual intensities and the vertical bars show the resolutions at which ice rings may be found. The red bars refer to suspected ice rings found by *AUSPEX*.

background model algorithm, the ice rings seem to have very little effect on the moments. For data processed with both the default background algorithm and the new global background model algorithm, the moments follow the expected theoretical curve. It is clear that the moments are not always a clear indicator of ice rings in the data. In particular, a mild pathology may not alter the moments such that the effect is visible through the noise; however, the effect may be visible for more prominent ice ring cases - such as for dataset 4PUC.

A plot of the R_{free} with resolution provides a better indication of the effect of ice rings in the data; however, it is only available after refinement. For datasets 4PUC and 4EF1 the effect of applying the new global background model algorithm is immediately clear: the R_{free} at ice ring resolutions is drastically decreased relative to the R_{free} using the default background algorithm. As also shown in the previous analysis, the difference observed in the R_{free} for dataset 4KW2 is negligible. Inspecting this plot may give some indication as to the effect of ice rings on the data, particularly for data containing very prominent ice rings - such as dataset 4PUC.

4.3.7 The effect of noise on the intensity moments

The observed reflection intensity, I_o is the sum of the true intensity, I_t, and a noise contribution, n, such that $I_o = I_t + n$. Therefore, the first and second moment of I_o can be written as:

$$\begin{aligned} &< I_o > =< I_t > + < n > \\ &< I_o^2 > =< I_o >^2 + var(I_t) + 2cov(I_t, n) + var(n). \end{aligned} \tag{4.13}$$

The normalised moment can then be written as the following ratio:

$$\frac{< I_o^2 >}{< I_o >^2} = 1 + \frac{var(I_t) + 2cov(I_t, n) + var(n)}{(< I_t > + < n >)^2}. \tag{4.14}$$

Let us denote $k = \frac{var(I_t)}{<I_t>^2}$, then the above ratio can be written as:

$$\frac{< I_o^2 >}{< I_o >^2} = 1 + \frac{k + 2cov(I_t, n)\frac{<n>}{<I_t>} + \frac{var(n)}{<I_t>^2}}{(1 + \frac{<n>}{<I_t>})^2}. \tag{4.15}$$

The value of k depends on the distribution of true intensities. For single crystal intensities without statistical peculiarities such as twinning and pseudo-translation, $k = 1$. For merohedral twinning, $k = \frac{1}{2}$. For n-fold twinning, $k = \frac{1}{n}$. Therefore, for single crystal, untwinned data, the ratio is given by:

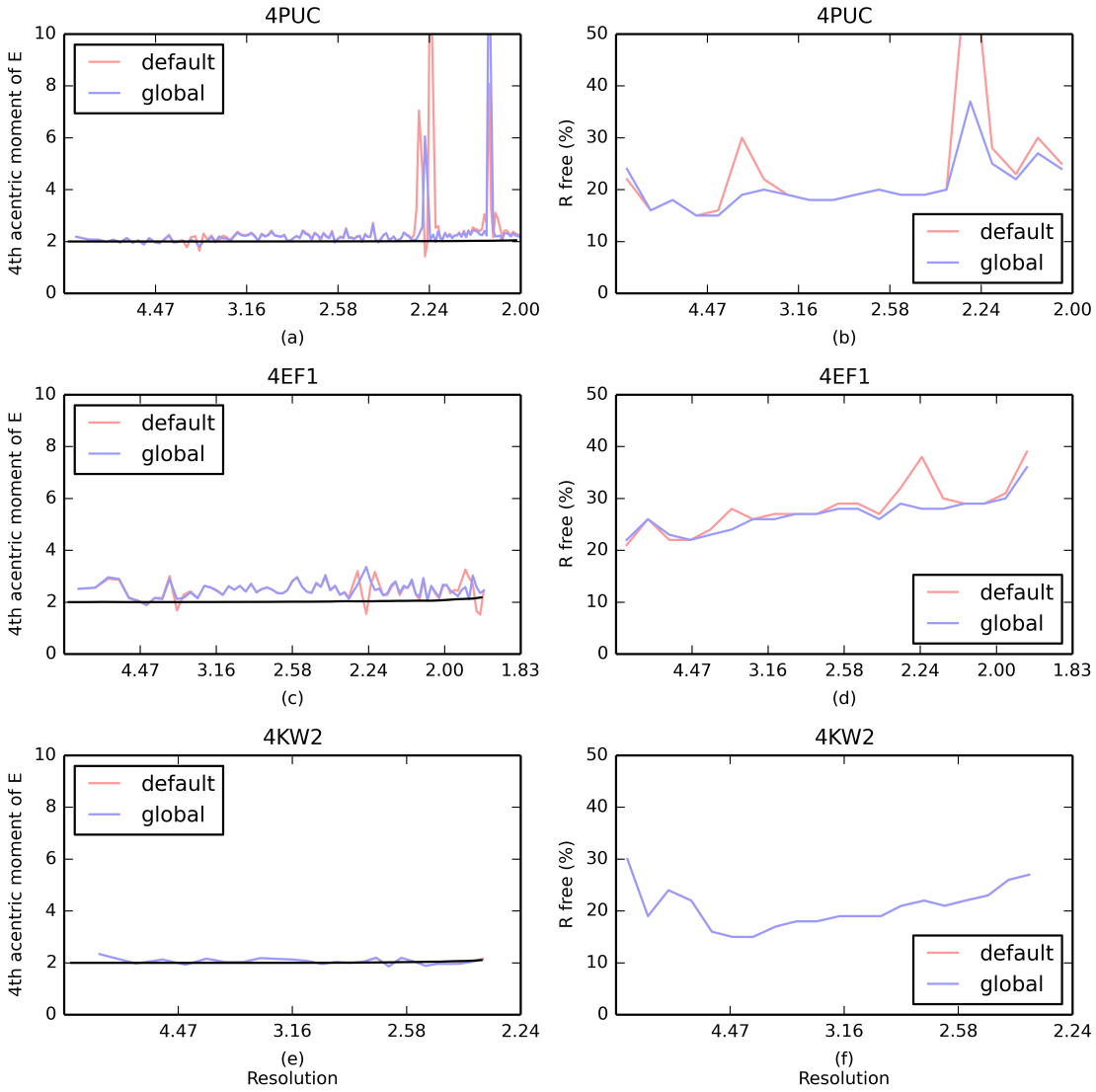


Figure 4.7: The 4th acentric moments of E, the normalised structure factors, *versus* resolution for datasets (a) 4PUC, (c) 4EF1 and (e) 4KW2. The red line indicates the default background algorithm and the blue light indicates the new global background model algorithm. The expected value for untwinned data is shown in black by the theoretical curve. The R_{free} versus resolution for datasets (b) 4PUC, (d) 4EF1 and (f) 4KW2. The red line indicates the default background algorithm and the blue light indicates the new global background model algorithm.

$$\frac{< I_o^2 >}{< I_o >^2} = 1 + \frac{1 + 2cov(I_t, n)\frac{<n>}{<I_t>} + \frac{var(n)}{<I_t>^2}}{(1 + \frac{<n>}{<I_t>})^2}. \tag{4.16}$$

Assuming in each case that there is no correlation between the signal and the noise (*i.e.* $\mathrm{cov}(I_t, n) = 0$).

1. If the mean noise, $< n >$ is 0 and the variance of the noise, $var(n)$, is close to zero, then:

$$\frac{< I_o^2 >}{< I_o >^2} = 2. \tag{4.17}$$

   Typically, at low resolution, where reflection intensities are large relative to the noise, the moments tend to this value.

2. If the mean noise, $< n >$ is non-zero and the variance of the noise, $var(n)$, is close to zero, then:

$$\frac{< I_o^2 >}{< I_o >^2} = 1 + \frac{1}{(1 + \frac{<n>}{<I_t>})^2}. \tag{4.18}$$

   In this case, the ratio depends on the value of the mean noise, $< n >$. If the reflections intensities are underestimated, then the mean noise will be negative. When the average noise is close to $- < I_t >$ then the ratio will become very large. If the reflection intensities are overestimated, then then mean noise is positive. As $< n > / < I_t >$ tends to infinity, the ratio will tend towards a value of 1, mimicking twinned data.

3. If the mean noise, $< n >$, is zero and the variance of the noise, $var(n)$, is non-zero, then:

$$\frac{< I_o^2 >}{< I_o >^2} = 2 + \frac{var(n)}{< I_t >^2} \tag{4.19}$$

   In this case, the ratio is always greater than 2 and as $var(n)$ increases relative to the true reflection intensity, $< I_t >$, the ratio also increases. At high resolution, where reflection intensities are small relative to the noise, this trend in the ratio towards infinity is often seen.

4.3.8 Application to data with no ice rings

As a control, a weak Thaumatin dataset collected on Diamond beamline I04 and known to contain no visible ice ring pathologies (Winter and Hall, 2014) was processed

to ensure that the new global background model algorithm gives good results in the case of well collected data. The average background over all resolution ranges is less than 1 count per pixel; it also has a low incidence of outliers in the background pixels. The dataset was processed to a resolution of 1.2 Å using the same procedure as described in Appendix A.3.

It was found that use of the global background model algorithm for this dataset resulted in no difference in the refinement R factors. Refinement with *REFMAC5* resulting in the same R_{free} for data processed with both the default background algorithm and the new global background model algorithm. When summation integration was used, the R_{free} was 18.1% in each case; when profile fitting was used, the R_{free} was 17.4% in each case. Thereby further illustrating the trend seen previously where a reduction is R_{free} is observed when using profile fitting over summation integration. Furthermore, plots of the moments with resolution and the R_{free} with resolution showed no difference between data processed with the default background algorithm and the global background model algorithm.

4.4 Conclusion

The use of a new global background model algorithm for the processing of X-ray diffraction data in the presence of ice rings has been presented. Traditional approaches to background modelling as implemented in current integration programs do not adequately cope with the task of modelling reflection background that is not well described by either a constant or a plane with a small slope. Consequently, these methods introduce systematic bias into the background estimation for reflections whose integration shoeboxes overlap with ice rings. This bias renders the majority of reflection intensities at certain resolutions unreliable if the dataset is contaminated by ice diffraction. At the peak of an ice ring, reflection intensities tend to be over-estimated due to an under-estimation of the reflection background. To either side of the ice ring, reflection intensities tend to be under-estimated due to an over-estimation of the reflection background. The use of a simple global background model algorithm has been shown to correct for these issues. Modelling the background in the presence of ice rings is challenging; however, correct modelling can have a noticeable effect on the downstream data processing. Finally, it is important to note that, whilst it is possible to correct for the effect of ice rings in data processing software, better results can be obtained by ensuring that samples are not contaminated with ice to begin with. This work was published in Parkhurst *et al.* (2017).

4.4.1 Future improvements

The current implementation uses a simple static background model which is applied to each image in the dataset. A future enhancement to the algorithm may be to employ a *scan-varying* global background model, where the model is allowed to vary over the course of the rotation scan. Additionally, the algorithm may be enhanced by generating a number of models (for example a flat and planar model as well as a curved model based on the global background model) and fit to each reflection with the model then being selected by a model selection algorithm, for example, the Akaike Information Criteria (AIC) (Akaike, 1973). Finally, the uncertainty in the determination of the global background model, as well as the uncertainty in the fitted background could be propagated to provide better estimates of the total error in the estimated intensities.

Chapter 5

Profile modelling for serial synchrotron X-ray diffraction data

5.1 Introduction

A major difficulty often faced by crystallographers is growing suitably large and well diffracting crystals to collect data to the desired resolution in a single crystal X-ray diffraction experiment. The diffracted signal is proportional to the illuminated volume of the crystal; therefore, for small crystals, in order to achieve the same strength of diffraction as from a larger crystal, a higher intensity beam is needed; this then entails more radiation damage to the crystal. Whilst cryo-cooling can help to slow the rate of radiation damage (Garman and Owen, 2006), it is often not enough to collect a full dataset from a single micro-crystal. Averaging intensity observations can help to improve the signal to noise ratio; therefore, when a large crystal is not available it is common to use data from many small crystals and to merge the resulting intensity measurements.

In recent years, the emergence and development of X-ray free electron lasers (XFEL) has popularised an experimental technique known as serial femtosecond crystallography (SFX) (Chapman *et al.*, 2011); the XFEL beam delivers a femtosecond pulse of X-rays which allows a diffraction pattern to be collected at room temperature without radiation damage - so called "diffraction before destruction". Since the XFEL beam destroys the crystal in a single pulse, only one diffraction pattern, a "still" image representing a single slice through reciprocal space, can be collected per crystal; therefore, in order to collect a complete dataset, many thousands of crystals are required.

At the same time, developments at synchrotron facilities have enabled MX beamlines to achieve micrometer sized X-ray beams that can match the size of very small crystals (Evans *et al.*, 2011). Together with the development of high viscosity

Lipidic cubic phase (LCP) injectors (Weierstall *et al.*, 2014), crystallographic sample chips (Mueller *et al.*, 2015) and fast, portable fixed-target acquisition systems (Sherrell *et al.*, 2015), this has enabled the experimental techniques used at XFEL facilities to be adapted for use at a synchrotron; a development known as high throughput serial synchrotron crystallography (SSX) (Stellato *et al.*, 2014; Gati *et al.*, 2014; Owen *et al.*, 2017; Weinert *et al.*, 2017). In the context of synchrotron experiments, the term "serial crystallography" takes on a slightly broader meaning, encompassing both "still" diffraction images, as in XFEL experiments, and individual small rotation images. Due to the longer exposure times used at synchrotron beamlines compared to XFEL beamlines, it is not possible to avoid radiation damage entirely; however, it can be partially alleviated by dividing the total dose over a large number of crystals collected with a small dose.

When performing a SSX experiment, it may be generally preferable to collect individual small rotations rather than still images since small rotations allow greater coverage of reciprocal space and the measurement of fully recorded reflections (Hasegawa *et al.*, 2017); however, as discussed below, there are cases when it is preferable to collect still images. The throughput for the collection of a serial dataset composed of many still images can be an order of magnitude higher than for the collection of a serial dataset composed of many small rotation images using the same translation hardware (Wierman *et al.*, 2019). For an experiment done using a micro-crystal chip on Diamond beamline I24, as shown in Figure 5.1, a full chip with 25,600 positions[1] can be collected in less than 10 minutes and approximately 25 grids can be collected every 12 hours: this is the equivalent of 640,000 positions shot, each of which may contain one or more crystals (Owen *et al.*, 2017).

For some cases, it is desirable and convenient to be able to perform the same experiment on both a synchrotron beamline and an XFEL beamline. For example, still image SSX experiments enable incremental dose experiments, where each position on the chip is exposed multiple times, to be performed in a time efficient manner; this enables dynamic studies of X-ray driven biological processes where electrons produced by the incident X-ray beam trigger reactions in the biological sample. Ebrahim *et al.* (2019) used a fixed target setup to perform a direct comparison of dose resolved SSX and radiation damage free XFEL structures of a radiation sensitive protein. Furthermore, synchrotron beam time is much easier to obtain than XFEL beam time; performing the SSX experiment at a synchrotron would allow for the assessment of sample and experimental workflows and could allow the feasibility of experiments for XFEL to be tested to provide useful supporting evidence for XFEL beamtime applications. The portable, compact hardware used for still image SSX

[1]Owen *et al.* (2017) refers to a previous version of the chip; information about the latest version can be found at https://www.diamond.ac.uk/Instruments/Mx/I24/I24-serial.html.

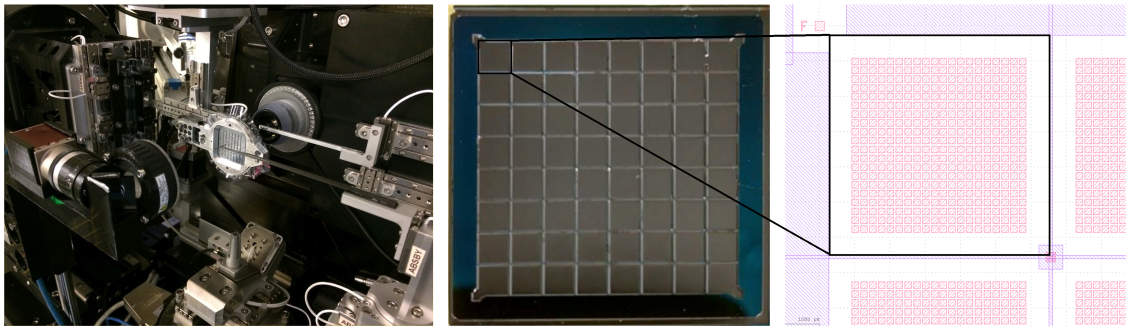


Figure 5.1: (a) The serial crystallography experimental setup on Diamond beamline I24, (b) a sample grid for data collection, and (c) a schematic of an element of the sample grid. Both traditional rotation experiments and serial crystallography experiments with either still images or small rotations can be performed on the beamline. Images courtesy of Danny Axford, Diamond Light Source.

enables the same experimental setup to be used in both cases (Sherrell *et al.*, 2015).

In order to cope with the specific challenges posed by still image X-ray diffraction data, new integration programs such as *cctbx.xfel* (Brewster *et al.*, 2016), *Cppxfel* (Ginn *et al.*, 2016) and *CrystFEL* (White *et al.*, 2016) have been developed, and traditional integration programs, previously optimised for processing data from rotation experiments, such as *XDS* (Kabsch, 2014) and *EVAL15* (Kroon-Batenburg *et al.*, 2015) have required some modification. Within *DIALS*, processing of still images is done *via* the *dials.stills_process* pipeline which shares algorithms and software with *cctbx.xfel*.

The challenges in processing still image X-ray diffraction data are numerous and related to the fact that each image only represents a single, thin slice through reciprocal space. Each still image typically only contains a small number of strong spots and, since each image contains diffraction from a different crystal with a unique unit cell and orientation, each image needs to be indexed independently. This usually requires the space group and approximate unit cell to be known *a priori* since a single still image will not, in general, provide enough information to determine them automatically. The small number of strong spots are then used to refine the experimental geometry. Although not an issue in SSX, in XFEL data processing a significant issue has been handling the complex metrology of the detectors used (Brewster *et al.*, 2018). If the detector is not moved during a data collection, the detector can be treated as fixed for all images; by performing a joint refinement of the detector parameters using data from all images, the detector position and orientation can be determined very accurately which consequently improves the determination of the crystal unit cell and orientation (Waterman *et al.*, 2016).

However, in the refinement of the experimental geometry from still image diffrac-

tion data, there is a dependence in the prediction on the exact form of the reflection reciprocal space profile model. Within *DIALS*, the default profile model has two components (Sauter *et al.*, 2014); the mosaic block size which manifests geometrically as spherical reciprocal lattice points, and an angular spread of mosaic blocks which manifests as a spherical cap around each reciprocal lattice vector (Nave, 1998). For still image diffraction data, the shape of the reciprocal lattice point (RLP) distribution has a substantial effect on the observed position of the reflection on the detector, whereas this is generally not the case for data collected using the rotation method. This can be seen by considering a reflection with a non-isotropic RLP distribution, centred on the RLP, at a random orientation with respect to the Ewald sphere. For a fully recorded reflection in a rotation experiment, the entire reflection will be rotated through the Ewald sphere and will be recorded on the detector. The intensity weighted centre of mass of the observed reflection on the detector images will then correspond to the predicted position of the reflection. For a still image, where only a slice of the RLP distribution results in diffraction, the intensity weighted centre of mass of the reflection recorded on the detector image will not, in general, correspond to the predicted position of the reflection. The predicted position, depends on the profile model used as shown in Figure 5.4. For a profile model consisting of a spherical RLP distribution with a monochromatic beam, the predicted centre of mass of the reflection is found by computing the diffracted beam vector corresponding to the point on the Ewald sphere closest to the RLP. For a profile model consisting of a spherical cap mosaicity model and a monochromatic beam, the predicted centre of mass of the reflection is found by computing the shortest rotation about an arbitrary axis that puts the reflection on the Ewald sphere (Sauter *et al.*, 2014). For a non-isotropic RLP distribution and monochromatic beam, the centre of mass is offset and is dependent on the shape of the profile. For a wavelength distribution with a wide bandpass, the centre of mass of the reflection tends towards the centre of mass of the RLP.

For still image diffraction data, the set of reflections which will actually be recorded on the diffraction image, and the set of reflections which will be entirely absent, are also dependent on the reflection profile model. Since an image represents a slice through reciprocal space and individual reciprocal lattice points are unlikely to be touching the surface of the Ewald sphere, the set of observed reflections will depend on the profile model. Given a RLP whose centre is some distance from the Ewald sphere, a selection criterion must be used to determine whether that reflection will be observed. A good algorithm will predict the reflections that are actually observed on the image without predicting reflections that are entirely absent. If observed reflections are not predicted, this is termed "under-prediction"; if absent reflections are predicted, this is termed "over-prediction".

In practice, for real data, it may not be obvious whether a reflection is not visibly recorded on the diffraction image because (a) no part of the reflection is in the diffracting condition or (b) it has an intrinsically low intensity. This can only be properly determined once the true intensities are known after scaling and merging of the data. Likewise, diffraction images may contain reflections from multiple lattices and contain other sources of noise. However, if a reflection indexed by a particular crystal is observed on the image and is not predicted then it is obvious that the set of reflections has been under-predicted by the algorithm. In the case of over-prediction, absent reflections can be dealt with at a later stage during scaling and merging of the data; however, for under-prediction, valuable information is lost completely after integration. Therefore, a small amount of over-prediction may be desirable to reduce loss of information. It has been observed that under-prediction of observed reflections is a particular problem for the standard prediction algorithm for still image diffraction data in *DIALS* (Winter, 2018; Axford, 2018; Nakane, 2018).

In this chapter, an enhanced model for the observed reflection profile is described; the model consists of two components: a Multivariate Normal distribution (MVN) is used to describe the distribution of reciprocal lattice vectors around the RLP for each reflection, and a Normal distribution is used to describe the distribution of wavelengths. By using a MVN distribution to describe the RLP distribution, non-isotropic spot shapes can be easily described. Additionally, Normal distributions have various useful properties which allow the parameters of the profile model to be estimated easily from the data *via* a maximum likelihood algorithm. It is shown that, application of the profile model to simulated data and SSX data collected from Diamond beamline I24 results in better determination of the crystal unit cell parameters and orientation, and allows more accurate prediction of the reflection positions on the detector. Furthermore, the enhanced algorithm is better able to predict which reflections are observed on the detector. The algorithm is implemented within a stand-alone program, *dials.potato*.

5.2 Algorithm

5.2.1 Model

In order to model the observed profile of the diffraction spots recorded on the detector, the finite shape of the RLP distribution and the distribution of wavelengths are considered. For convenience, the mathematical symbols used along with their definitions are given in the list of symbols on page 13.

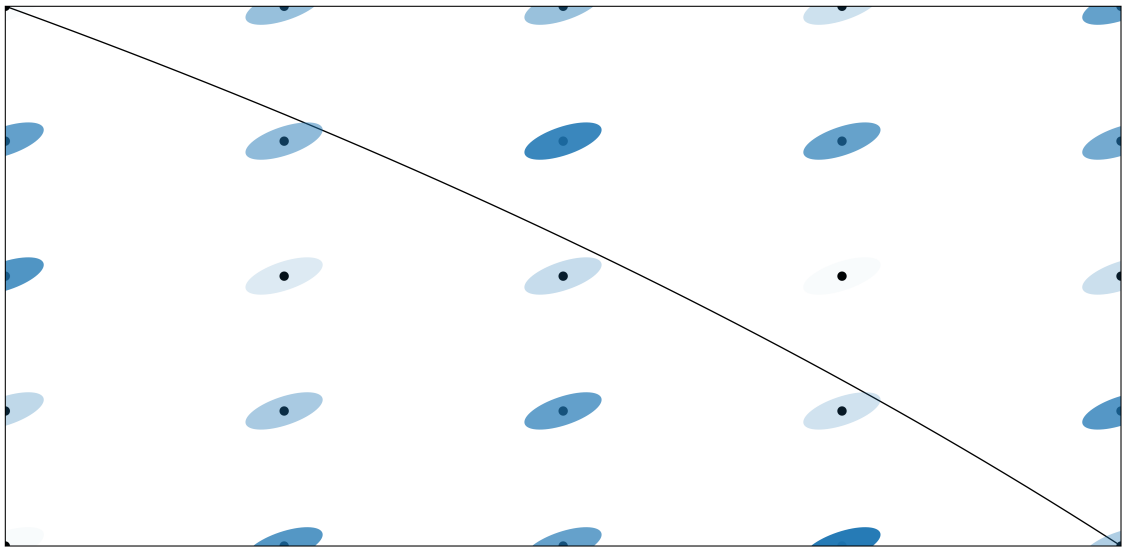


Figure 5.2: The RLP distribution in reciprocal space. The size of each spot in reciprocal space is described by a MVN distribution. The black line represents the surface of the Ewald sphere in the case of a δ-function model. The distribution of reciprocal lattice vectors around each RLP is the same; however, it is worth noting that the intensity of each reflection will be different and this is indicated in the diagram by a variation in the colours of the reflection profiles.

Reciprocal lattice point distribution

The RLP distribution can be modelled as a Multivariate Normal (MVN) distribution in reciprocal space centred on the reciprocal lattice point and described by a 3D covariance matrix. The MVN distribution allows for anisotropy in the spot shape and has various properties that simplify calculations and make it a convenient choice for the RLP distribution. For a RLP, $\boldsymbol{r}_0 = \boldsymbol{UBh}$, the distribution of reciprocal lattice vectors, $\boldsymbol{r}$, is given by

$$\boldsymbol{r} \sim \mathcal{N}(\boldsymbol{r}_0, \boldsymbol{M}). \tag{5.1}$$

Where the RLP covariance matrix, $\boldsymbol{M}$, in reciprocal space coordinates has the following components:

$$\boldsymbol{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix}. \tag{5.2}$$

All reciprocal lattice points are assumed to have the same distribution in reciprocal space so the shapes of all spots can be described by a single set of model parameters as shown in Figure 5.2.

Wavelength distribution

The distribution of X-ray wavelengths is modelled as a Normal distribution with mean wavelength, λ_0, and variance, σ_λ^2, such that:

$$\lambda \sim \mathcal{N}(\lambda_0, \sigma_\lambda^2). \tag{5.3}$$

If the variance of the wavelength distribution is equal to zero, the distribution is modelled as a δ-function, *i.e.* a monochromatic beam with wavelength λ_0; in this case, simplifying approximations can be used for the general impact of the reflection. Synchrotron beamlines for MX are typically monochromatic with very small wavelength dispersion (the energy resolution is $\mathrm{d}E/E \approx 1 \times 10^{-4}$); in this case, a δ-function wavelength distribution may be sufficient as an approximation. However, there are cases, such as pink beam experiments, where a spread of wavelengths needs to be considered. Considering the spread of wavelengths to be approximately Normally distributed allows for convenient modelling of the product of the reciprocal lattice point profile with the wavelength distribution as shown in Section 5.2.3. This model is appropriate for data collected at a synchrotron; however, X-rays from XFEL beamlines typically have a multi-modal peaked wavelength distribution that is different for each image and is not well characterised by a smoothly varying unimodal distribution.

Reflection specific coordinate system

It is convenient to perform some calculations in a reflection specific coordinate system shown in Figure 5.3. Given a reciprocal lattice vector, $\boldsymbol{r}_0$, and an incident beam vector, $\boldsymbol{s}_0$ with length $|\boldsymbol{s}_0| = 1/\lambda_0$, the vector to the reciprocal lattice point in laboratory space is $\boldsymbol{s}_2 = \boldsymbol{s}_0 + \boldsymbol{r}_0$. If $|\boldsymbol{s}_2| = |\boldsymbol{s}_0|$ then the centre of the RLP lies directly on the Ewald sphere; however, this is not generally the case. For a RLP with laboratory vector, $\boldsymbol{s}_2$, the basis vectors of the reflection specific coordinate system are defined as follows:

$$\begin{aligned} \boldsymbol{e}_1 &= \frac{\boldsymbol{s}_2 \times \boldsymbol{s}_0}{|\boldsymbol{s}_2 \times \boldsymbol{s}_0|} \\ \boldsymbol{e}_2 &= \frac{\boldsymbol{s}_2 \times \boldsymbol{e}_1}{|\boldsymbol{s}_2 \times \boldsymbol{e}_1|} \\ \boldsymbol{e}_3 &= \frac{\boldsymbol{s}_2}{|\boldsymbol{s}_2|} \end{aligned} \tag{5.4}$$

The reflection specific coordinate system is similar to that defined in Kabsch (2010a) for describing the standard reflection profiles. The $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ vectors form a tangent plane on the Ewald sphere surface and describe the transverse and radial

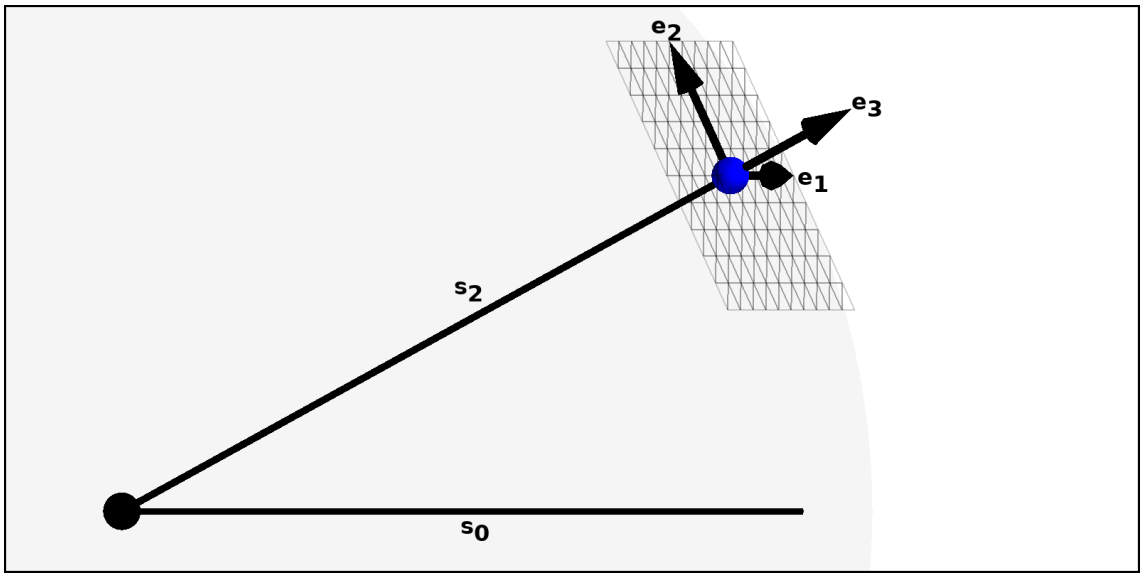


Figure 5.3: The reflection specific coordinate system.

components of the spot shape around the beam centre. The $\boldsymbol{e}_3$ vector is just the unit vector pointing from the centre of the Ewald sphere to the centre of the reciprocal lattice point. These basis vectors form a rotation matrix:

$$\boldsymbol{R}_e = \begin{pmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33}. \end{pmatrix} \tag{5.5}$$

This rotation matrix transforms the RLP laboratory space vector, $\boldsymbol{s}_2$, to lie along the Z axis of the local reflection coordinate system.

Model parametrisation

The RLP distribution is described by its 3D covariance matrix. In order to be valid, the covariance matrix is required to be positive semi-definite. This can be enforced by using the Cholesky decomposition to parameterise the RLP covariance matrix. The parametrisation is given such that $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^*$, where $\boldsymbol{L}^*$ is the conjugate transpose of $\boldsymbol{L}$, a lower triangular matrix with positive diagonal elements. The RLP covariance matrix is then fully described by 6 parameters, $(m_1, m_2, m_3, m_4, m_5, m_6)$, the non-zero elements of matrix $\boldsymbol{L}$, where

$$\boldsymbol{L} = \begin{pmatrix} m_1 & 0 & 0 \\ m_2 & m_3 & 0 \\ m_4 & m_5 & m_6 \end{pmatrix}. \tag{5.6}$$

The RLP covariance matrix is then:

$$\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^* = \begin{pmatrix} m_1^2 & m_2 m_1 & m_4 m_1 \\ m_2 m_1 & m_2^2 + m_3^2 & m_4 m_2 + m_5 m_3 \\ m_4 m_1 & m_4 m_2 + m_5 m_3 & m_4^2 + m_5^2 + m_6^2 \end{pmatrix}. \tag{5.7}$$

For the wavelength distribution, if the δ-function wavelength model is used, no additional parameters are required. If a Normal distribution is used, a single parameter is required to describe the variance, σ_λ^2.

5.2.2 General impact for δ-function wavelength model

The general impact discussed here refers to the distribution of diffracted beam vectors on a virtual spherical detector at a distance $|\boldsymbol{s}_0|$; *i.e.* the distribution of diffracted beam vectors on the Ewald sphere. This distribution can be described simply, whereas the distribution projected onto the flat detector plane will be necessarily more complex. It is assumed that the Ewald sphere is flat on the scale of the size of a spot. In this case, the general impact is approximated by the shape of the spot on a tangent plane on the Ewald sphere surface as given by the local reflection specific coordinate system.

Given the RLP distribution covariance matrix, $\boldsymbol{M}$, the diffracted beam vector, $\boldsymbol{s}_2$ and the local reflection specific coordinate system transformation matrix, $\boldsymbol{R}_e$, the shape of the reflection profile in the local reflection specific coordinate system is simply the transformed RLP distribution which is a MVN distribution with mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, given by:

$$\begin{aligned} \boldsymbol{\mu} &= \boldsymbol{R}_e \boldsymbol{s}_2 \\ \boldsymbol{\Sigma} &= \boldsymbol{R}_e \boldsymbol{M} \boldsymbol{R}_e{}^T. \end{aligned} \tag{5.8}$$

To compute the general impact then requires the calculation of the conditional distribution of the MVN distribution on the tangent plane of the Ewald sphere. This requires the decomposition of the 3D MVN distribution, $P(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, into the 1D marginal distribution, $P(\boldsymbol{e}_3)$, along the $\boldsymbol{e}_3$ basis vector, and the 2D conditional distribution, $P(\boldsymbol{e}_1, \boldsymbol{e}_2|\boldsymbol{e}_3)$, on the plane formed by the $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ basis vectors conditional on the distance between the Ewald sphere and RLP along the $\boldsymbol{e}_3$ basis vector. The transformed RLP distribution, $P(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, can then be written as $P(\boldsymbol{e}_1, \boldsymbol{e}_2|\boldsymbol{e}_3)P(\boldsymbol{e}_3)$. In this case, the marginal distribution is the projection of the MVN distribution onto the $\boldsymbol{e}_3$ basis vector and the conditional distribution is a slice of the MVN distribution on the $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ plane at a given position along $\boldsymbol{e}_3$. A convenient property of the MVN distribution is that both the marginal and conditional distributions are also Normal distributions. Given a MVN distribution with mean vector, $\boldsymbol{\mu}$, and covariance matrix,

$\boldsymbol{\Sigma}$, the marginal and conditional distributions can be computed by partitioning the mean vector and covariance matrix such that:

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \tag{5.9}$$

and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}. \tag{5.10}$$

Here, $\boldsymbol{\mu}_1$ is the 2D vector mean of the conditional distribution, $\boldsymbol{\mu}_2$ is the scalar mean of the marginal distribution, $\boldsymbol{\Sigma}_{11}$ is the 2D covariance matrix of the conditional distribution and $\boldsymbol{\Sigma}_{22}$ is the scalar variance of the marginal distribution. The marginal distribution has a mean, $\tilde{\mu}$, and variance, $\tilde{\Sigma}$, given by:

$$\begin{aligned} \tilde{\mu} &= \boldsymbol{\mu}_2 = \boldsymbol{\mu}_Z \\ \tilde{\Sigma} &= \boldsymbol{\Sigma}_{22} = \boldsymbol{\Sigma}_Z. \end{aligned} \tag{5.11}$$

Using the properties of block matrix inversions, as shown in Appendix A.4, the conditional distribution of the MVN distribution can be shown to have mean, $\bar{\boldsymbol{\mu}}$, and covariance matrix, $\bar{\boldsymbol{\Sigma}}$, given by:

$$\begin{aligned} \bar{\boldsymbol{\mu}} &= \boldsymbol{\mu}_1 + (\boldsymbol{\Sigma}_{12})(\boldsymbol{\Sigma}_{22})^{-1}(|\boldsymbol{s}_0| - \boldsymbol{\mu}_2) = \boldsymbol{\mu}_{XY} \\ \bar{\boldsymbol{\Sigma}} &= \boldsymbol{\Sigma}_{11} - (\boldsymbol{\Sigma}_{12})(\boldsymbol{\Sigma}_{22})^{-1}(\boldsymbol{\Sigma}_{21}) = \boldsymbol{\Sigma}_{XY}. \end{aligned} \tag{5.12}$$

The marginal distribution gives information about the distribution of spots that will be observed and the conditional distribution gives information about the general impact of a spot in its local reflection specific coordinate system. The mean of the conditional distribution can be understood as the "central impact" and the variance of the conditional distribution can be understood as describing the "general impact". The centre of mass of the reflection recorded on the Ewald sphere can then be approximated by using the point in the reflection specific coordinate system at the mean of the conditional distribution on the Ewald sphere and rotating back into the laboratory frame as follows and shown in Figure 5.4:

$$\boldsymbol{s}_1 = \boldsymbol{R}_e{}^T \begin{pmatrix} \boldsymbol{\mu}_{XY} \\ |\boldsymbol{s}_0| \end{pmatrix} \tag{5.13}$$

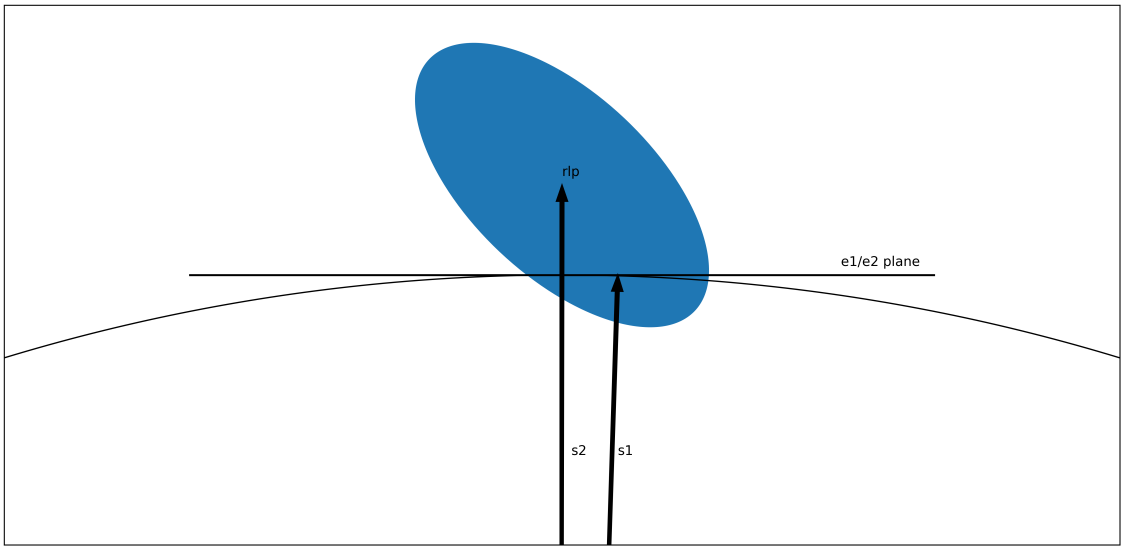


Figure 5.4: The predicted diffracted beam vector; the mean of the conditional distribution on the Ewald sphere approximated by the plane formed by the $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ axes in the reflection specific coordinate system.

5.2.3 General impact for Normal wavelength model

For the Normally distributed wavelength model, the calculation of the general impact is slightly more complicated than for the monochromatic case. The interaction between the wavelength distribution and the reciprocal lattice point distribution, for an instantaneous still image, can be thought of as the product of two distributions in reciprocal space: the RLP distribution and the distribution of Ewald spheres resulting from the wavelength spread as shown in Figure 5.5. In general, this product distribution will be very complicated and may not be possible to describe analytically. However, it is possible to describe this interaction in terms of a product of Normal distributions by considering a linear approximation.

Ewald sphere distribution

A Normal distribution of wavelengths can be visualised as a distribution of Ewald spheres as shown in Figure 5.6. In order to model the interaction of the Ewald sphere distribution and the RLP distribution for a given reflection, an approximation of the Ewald sphere distribution in the local reflection specific coordinate system is required. To this end, the Ewald sphere distribution is modelled for each reflection as a Normal distribution along the $\boldsymbol{e}_3$ basis vector of the local reflection specific coordinate system. For a reciprocal lattice vector, $\boldsymbol{r}$, to satisfy the diffraction condition at wavelength, λ, the following must hold:

$$\left| \frac{\boldsymbol{s}_0}{\lambda |\boldsymbol{s}_0|} + \boldsymbol{r} \right| = \left| \frac{\boldsymbol{s}_0}{\lambda |\boldsymbol{s}_0|} \right|. \tag{5.14}$$

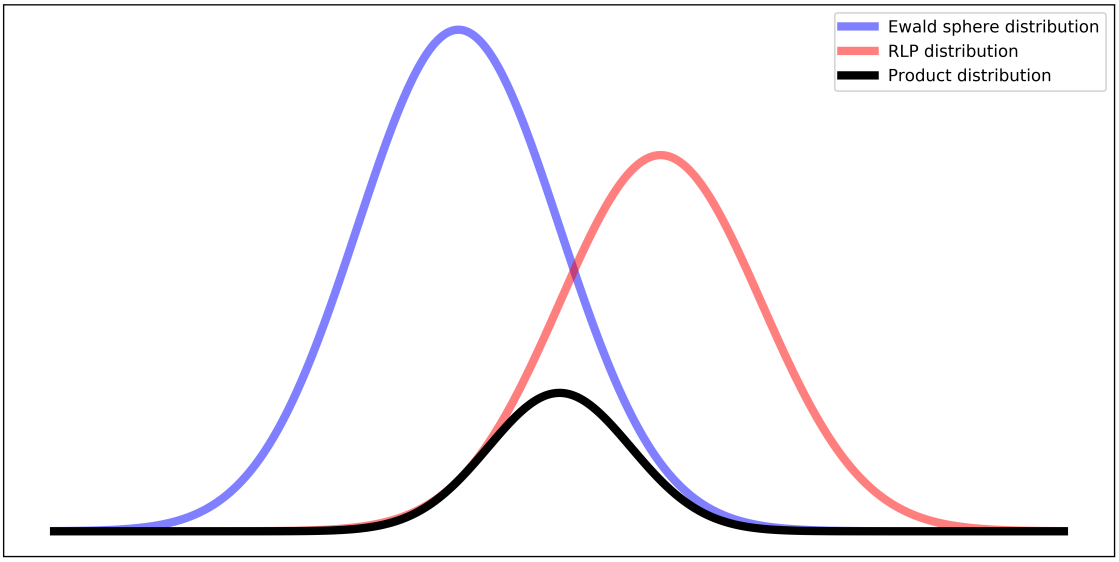


Figure 5.5: The product of the Ewald sphere distribution with the reciprocal lattice point distribution.

Therefore, the wavelength that will excite a particular reciprocal lattice point, $\boldsymbol{r}$ is given by:

$$\lambda = -\frac{2}{|\boldsymbol{s}_0|}\frac{\boldsymbol{s}_0 \cdot \boldsymbol{r}}{\boldsymbol{r} \cdot \boldsymbol{r}}. \tag{5.15}$$

Given a line between a point in reciprocal space, $\boldsymbol{r}_E = \frac{\boldsymbol{s}_2}{|\boldsymbol{s}_2|} - \boldsymbol{s}_0$, lying on the mean Ewald sphere with radius, $|\boldsymbol{s}_0|$, and the origin, points in reciprocal space, $\boldsymbol{r}$, lying along that line can be written as a function of z along that line as follows:

$$\boldsymbol{r} = z\left(\frac{\boldsymbol{s}_0 + \boldsymbol{r}_E}{|\boldsymbol{s}_0|}\right) - \boldsymbol{s}_0. \tag{5.16}$$

By taking $\boldsymbol{r}$ as a function of z, the Taylor expansion of Equation 5.15 around $z = |\boldsymbol{s}_0|$ is

$$\lambda(z) \approx \frac{1}{|\boldsymbol{s}_0|} - \frac{2(z - |\boldsymbol{s}_0|)}{\boldsymbol{r}_E \cdot \boldsymbol{r}_E}. \tag{5.17}$$

Rearranging for z then gives:

$$z = -\frac{(\lambda - \lambda_0)\boldsymbol{r}_E \cdot \boldsymbol{r}_E}{2} + \frac{1}{\lambda_0} \tag{5.18}$$

If the wavelength is Normally distributed with mean, λ_0, and variance, σ_λ^2, then z is Normally distributed about $|\boldsymbol{s}_0|$ with variance given by:

$$\sigma_E^2 = \left(\frac{\boldsymbol{r}_E \cdot \boldsymbol{r}_E}{2}\right)^2 \sigma_\lambda^2. \tag{5.19}$$

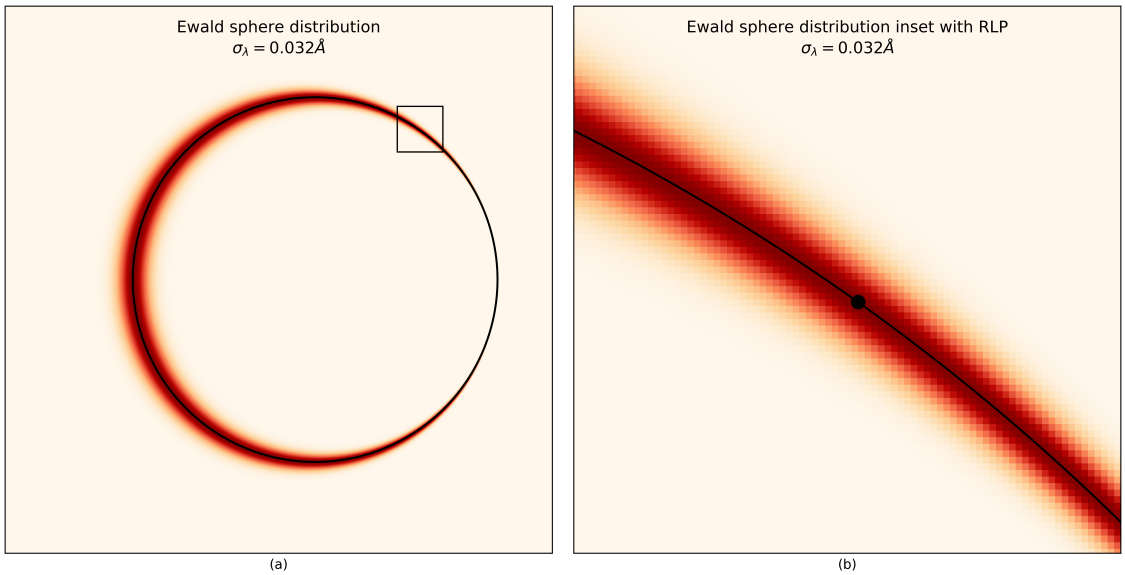


Figure 5.6: The distribution of Ewald spheres resulting from a Normally distributed spread in wavelengths (a). The inset rectangle is shown in (b) with a RLP. Even for a large spread in wavelengths, the Ewald sphere appears of uniform thickness on the scale on a RLP.

The spread of Ewald spheres can be approximated as a Normal distribution around the mean Ewald sphere where the variance of the distribution increases with half the squared distance to the point on the Ewald sphere in reciprocal space.

Product of two Normal probability density functions

The product of two Normal probability density functions, $\mathcal{N}(\mu_1, \sigma_1^2)$ and $\mathcal{N}(\mu_2, \sigma_2^2)$, is a scaled Normal distribution with mean and variance given by:

$$\begin{aligned} \mu &= \frac{\mu_1\sigma_2^2 + \mu_2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \\ \sigma^2 &= \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}. \end{aligned} \tag{5.20}$$

Product of the Ewald sphere and Reciprocal lattice point distributions

The local approximation of the product of the Ewald sphere distribution and the RLP distribution can be described by considering the conditional and marginal distributions of the RLP distribution. The effect of the wavelength distribution can be approximated by a Normal distribution along a vector from the centre of the mean Ewald sphere with radius $|\boldsymbol{s}_0|$. The distribution has a mean, $|\boldsymbol{s}_0|$, and variance σ_E^2. The marginal distribution of the RLP distribution along this vector has a mean, $\boldsymbol{\mu}_2$, and variance, $\boldsymbol{\Sigma}_{22}$. Using the result of the product of two Normal probability density functions as shown above, the product of the local Ewald sphere distribution

and the marginal distribution of the RLP distribution has a mean and variance given by

$$\begin{aligned} \boldsymbol{p}_2 &= \frac{\boldsymbol{\mu}_2\sigma_E^2 + |\boldsymbol{s}_0|\boldsymbol{\Sigma}_{22}}{\boldsymbol{\Sigma}_{22} + \sigma_E^2} \\ \boldsymbol{P}_{22} &= \frac{\sigma_E^2\boldsymbol{\Sigma}_{22}}{\boldsymbol{\Sigma}_{22} + \sigma_E^2} = \kappa\boldsymbol{\Sigma}_{22}. \end{aligned} \tag{5.21}$$

The Ewald sphere distribution has the effect of increasing the variance of the marginal distribution by a factor, $\kappa = \frac{\sigma_E^2}{\boldsymbol{\Sigma}_{22}+\sigma_E^2}$, and moving the mean of the marginal distribution towards the centre of the RLP. Using the properties of block matrices, as shown in Appendix A.5, the joint distribution of the conditional reciprocal lattice distribution and the product of the marginal and Ewald sphere distribution can be shown to have a mean, $\boldsymbol{p}$, and covariance matrix, $\boldsymbol{P}$, as follows:

$$\begin{aligned} \boldsymbol{p} &= \begin{pmatrix} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{p}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{p}_2 \end{pmatrix} \\ \boldsymbol{P} &= \begin{pmatrix} \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(1-\kappa) & \kappa\boldsymbol{\Sigma}_{12} \\ \kappa\boldsymbol{\Sigma}_{21} & \kappa\boldsymbol{\Sigma}_{22} \end{pmatrix}. \end{aligned} \tag{5.22}$$

This has the property that, as the variance of the wavelength distribution goes to zero, κ goes to zero and the product distribution tends to the conditional distribution of the RLP distribution. As the variance of the wavelength distribution goes to infinity, κ goes to 1 and the product distribution tends to the RLP distribution. This is shown graphically in Figure 5.7 which shows the product distribution for a monochromatic beam, small bandpass and large bandpass; the first column in the figure shows the local distribution of Ewald spheres; the second column shows the anisotropic RLP distribution; the third column shows the product of the Ewald sphere distribution and the RLP distribution; the fourth column shows a 2D slice of the general impact of the spot on the detector, the true location of the central impact using the profile model and naive estimate of the central impact.

Distribution of diffracted beam vectors

Given the shape of the illuminated spot in reciprocal space, let us now consider the distribution of diffracted beam vectors that would arise from the spot. The spread of wavelengths results in an increased spread in spot size on the detector. The diffracted beam vector for a point in reciprocal space, $\boldsymbol{r}$, can be written as follows, using Equation 5.15:

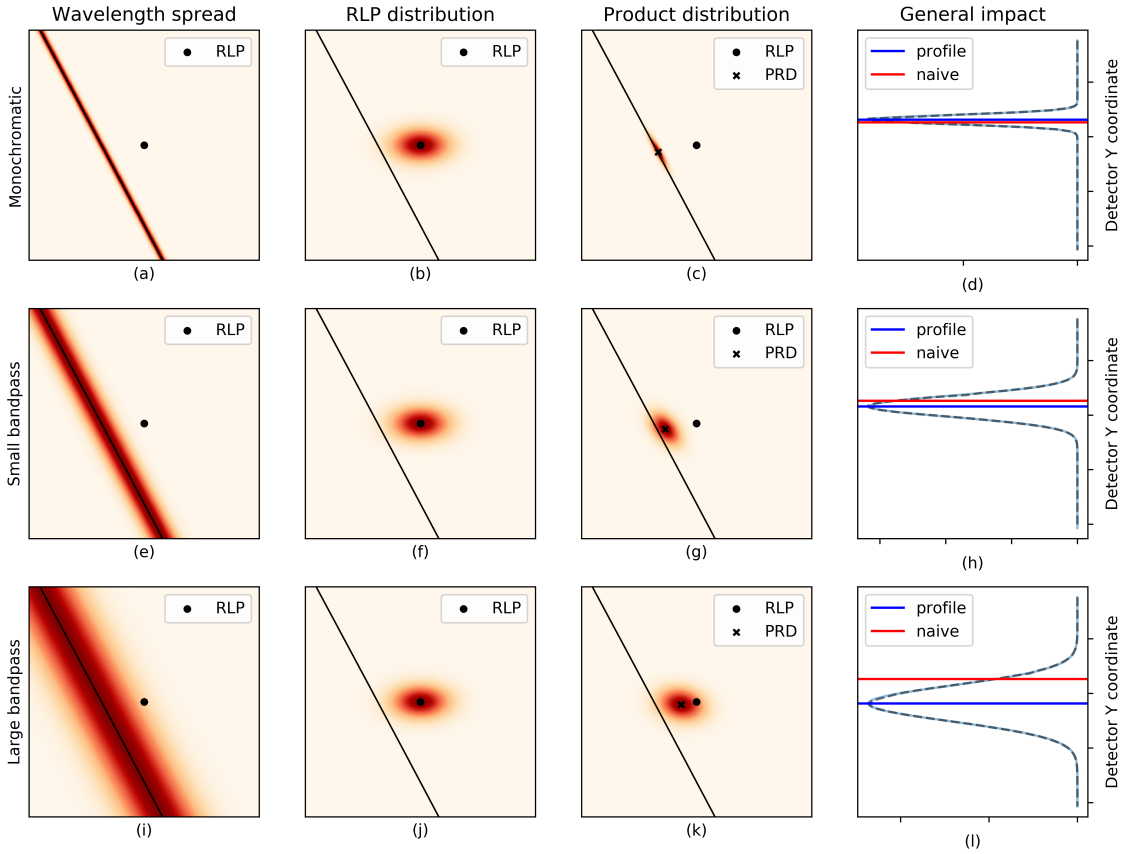


Figure 5.7: The distribution resulting from the interaction of the Ewald sphere distribution with the reciprocal lattice point distribution for a monochromatic beam, small bandpass and large bandpass. The centre of the reciprocal lattice point (RLP) is marked with a circle in the heat maps and the centre of mass of the product distribution (PRD) is marked with a cross. the resulting general impact is shown in each case along with the true central impact (profile) and the predicted central impact if the naive prediction is used (naive). The approximated general impact is shown as the solid line and the true general impact as the coincident dashed line.

$$\boldsymbol{s}(\boldsymbol{r}) = \frac{\boldsymbol{s}_0}{|\boldsymbol{s}_0|}\frac{1}{\lambda(\boldsymbol{r})} + \boldsymbol{r} = -\frac{1}{2}\frac{(\boldsymbol{r}\cdot\boldsymbol{r})}{(\boldsymbol{s}_0\cdot\boldsymbol{r})}\boldsymbol{s}_0 + \boldsymbol{r}. \tag{5.23}$$

Equation 5.23 can be approximated by performing the Taylor expansion about the central point of the distribution in reciprocal space, $\boldsymbol{r}_c = \boldsymbol{p}$, such that $\boldsymbol{s}(\boldsymbol{r}) \approx \boldsymbol{s}(\boldsymbol{r}_c) + \boldsymbol{J}(\boldsymbol{r}_c)(\boldsymbol{r} - \boldsymbol{r}_c)$. Here, the matrix of derivatives, $\boldsymbol{J}$ is given by:

$$\boldsymbol{J}(\boldsymbol{r}_c) = \mathbb{1} - \frac{1}{2}\boldsymbol{s}_0\left(\frac{2(\boldsymbol{r}_c\cdot\boldsymbol{s}_0)\boldsymbol{r}_c - (\boldsymbol{r}_c\cdot\boldsymbol{r}_c)\boldsymbol{s}_0}{(\boldsymbol{r}_c\cdot\boldsymbol{s}_0)^2}\right)^T. \tag{5.24}$$

The distribution of diffracted beam vectors can therefore be approximated by a Normal distribution with mean, $\boldsymbol{q} = \boldsymbol{s}(\boldsymbol{r}_c)$, and variance, $\boldsymbol{Q} = \boldsymbol{J}\boldsymbol{P}\boldsymbol{J}^T$. To compute the general impact on the virtual spherical detector, the distribution is marginalised in the local reflection specific coordinate system on the $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ plane. The mean and covariance matrix are partitioned as follows:

$$\begin{aligned}\boldsymbol{q} &= \begin{pmatrix}\boldsymbol{q}_1\\ \boldsymbol{q}_2\end{pmatrix}\\ \boldsymbol{Q} &= \begin{pmatrix}\boldsymbol{Q}_{11} & \boldsymbol{Q}_{12}\\ \boldsymbol{Q}_{21} & \boldsymbol{Q}_{22}\end{pmatrix}.\end{aligned} \tag{5.25}$$

The distribution of the general impact in the local reflection specific coordinate system is then a Normal distribution with mean, $\boldsymbol{\mu}_{XY} = \boldsymbol{q}_1$, and covariance matrix, $\boldsymbol{\Sigma}_{XY} = \boldsymbol{Q}_{11}$. The variance of the distribution around the mean Ewald sphere is $\boldsymbol{\Sigma}_Z = \kappa\boldsymbol{\Sigma}_{22}$.

5.2.4 Parameter estimation

Observed data

The input data to the algorithm is a set of strong indexed spots and an initial experimental model. Each strong spot contributes various pieces of information from which the model parameters may be inferred.

1. The observed centroid position gives information about where the centre of mass of the spot should be predicted.

2. The distribution of pixel values making up the spot gives information about the general impact of the spot and the size of the spot in reciprocal space.

3. The list of strong spots itself gives information about which spots are close to the Ewald sphere and the distribution around the Ewald sphere surface.

Maximum likelihood algorithm

Given a set of N observed strong spots, the parameters of the profile model can be estimated from the observed data *via* a maximum likelihood estimator. Each observed spot is recorded across a number of pixels on the detector; this constitutes the general impact of the spot on the detector. Rather than considering this distribution on the detector directly, the distribution of the pixels counts mapped to the local reflection specific coordinate system is used. For both the monochromatic and Normally distributed wavelength models, the distribution of the diffracted beam vectors in the local reflection specific coordinate system can be approximated as a bivariate Normal distribution. The projection of this distribution onto the detector surface results in a distribution that is in general not Normal; therefore, this approximation significantly simplifies the estimation of the model parameters.

The probability of observing a count in the reflection specific coordinate system can be given in terms of the conditional probability distribution on the tangent plane to the Ewald sphere surface along the $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ axes, and the marginal probability distribution on the $\boldsymbol{e}_3$ axis orthogonal to that plane, $P(x, y, z) = P(x, y|z)P(z)$. Each spot is predicted to be a distance, $\epsilon = |\boldsymbol{s}_2| - |\boldsymbol{s}_0|$ from the Ewald sphere along the $\boldsymbol{e}_3$ axis and each pixel in the spot maps to a point, $\boldsymbol{x} = (x, y)$, in the $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ plane. Given the mean, $\boldsymbol{\mu}_{XY}$, and covariance matrix, $\boldsymbol{\Sigma}_{XY}$, of the expected distribution of diffracted beam vectors, and the expected variance of the distribution of observed strong spots around the diffracting condition, $\boldsymbol{\Sigma}_Z$, the probability of observing a count at a point, $(\boldsymbol{x}, \epsilon)$ is

$$\begin{aligned} P(x, y|z) &= \frac{1}{\sqrt{(2\pi)^2|\boldsymbol{\Sigma}_{XY}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_{XY})^T {\boldsymbol{\Sigma}_{XY}}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{XY})\right) \\ P(z) &= \frac{1}{\sqrt{2\pi\boldsymbol{\Sigma}_Z}} \exp\left(-\frac{1}{2}\epsilon^2 {\boldsymbol{\Sigma}_Z}^{-1}\right). \end{aligned} \tag{5.26}$$

Therefore, given N observed strong spots, where a spot, i, has contributions from M_i detector pixels, j, and each pixel in the spot, (i, j), has observed counts of $c_{i,j}$, the log likelihood can be written as

$$\begin{aligned} \mathcal{L} = &-\frac{1}{2}\sum_i^N \sum_j^{M_i} c_{i,j}\left[\ln(\boldsymbol{\Sigma}_{Zi}) + {\boldsymbol{\Sigma}_Z}_i^{-1}\epsilon_i^2\right] \\ &-\frac{1}{2}\sum_i^N \sum_j^{M_i} c_{i,j}\left[\ln(|\boldsymbol{\Sigma}_{XYi}|) + (\boldsymbol{x}_{i,j} - \boldsymbol{\mu}_{XYi})^T {\boldsymbol{\Sigma}_{XY}}_i^{-1}(\boldsymbol{x}_{i,j} - \boldsymbol{\mu}_{XYi})\right]. \end{aligned} \tag{5.27}$$

The total observed counts, c_{tot}, mean, $\bar{\boldsymbol{x}}$, and covariance, $\boldsymbol{S}$, of each spot are given

by

$$c_{tot} = \sum_j^M c_j, \quad \bar{\boldsymbol{x}} = \frac{1}{c_{tot}} \sum_j^M c_j \boldsymbol{x}_j, \quad \boldsymbol{S} = \frac{1}{c_{tot}} \sum_j^M c_j (\boldsymbol{x}_j - \bar{\boldsymbol{x}})(\boldsymbol{x}_j - \bar{\boldsymbol{x}})^T. \tag{5.28}$$

By making use of the cyclic permutation property of the matrix trace operator, the log likelihood equation can then be written in terms of these statistics as follows

$$\begin{aligned} \mathcal{L} = &- \frac{1}{2} \sum_i^N c_{tot,i} \left[\ln(\boldsymbol{\Sigma}_{Zi}) + \boldsymbol{\Sigma}_{Z_i}^{-1} \epsilon_i^2 \right] \\ &- \frac{1}{2} \sum_i^N c_{tot,i} \left[\ln(|\boldsymbol{\Sigma}_{XYi}|) + \mathrm{tr}(\boldsymbol{\Sigma}_{XY_i}^{-1} \boldsymbol{S}_i) + (\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi})^T \boldsymbol{\Sigma}_{XY_i}^{-1} (\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi}) \right]. \end{aligned} \tag{5.29}$$

This removes the inner sum over the spot pixels from the likelihood equation. Since the statistics, $\bar{\boldsymbol{x}}$, c_{tot} and $\boldsymbol{S}$ do not depend on the model parameters, they only need to be calculated once; therefore, the pixel values only need to be accessed once rather than each time the log likelihood is evaluated. This simplification allows a practical implementation of the algorithm that is computationally less expensive.

To estimate the parameters, the Fisher scoring algorithm is used (Fisher, 1922). This algorithm is similar to the standard Newton Raphson algorithm for optimisation, except that the Fisher information matrix (the negative of the expected value of the Hessian) is used rather than the observed Hessian. Since the observed Hessian depends on the data, it can become poorly conditioned and the resulting optimisation can diverge. In contrast, the Fisher information matrix, $\mathcal{I}$, is guaranteed to be positive semi-definite (the value of $\boldsymbol{z}^T \mathcal{I} \boldsymbol{z}$ is non-negative for any non-zero column vector $\boldsymbol{z}$) so convergence is more robust. The Fisher scoring algorithm also has the benefit of only requiring the first derivatives to be explicitly calculated as shown in Equation 5.34; this results in a simpler algorithm and a less computationally intensive implementation (Osborne, 1992). At each iteration, $(t+1)$, the parameters $\boldsymbol{\beta}$ are updated according to the following equation

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathcal{I} \left(\boldsymbol{\beta}^{(t)} \right)^{-1} V \left(\boldsymbol{\beta}^{(t)} \right). \tag{5.30}$$

Where $V(\beta_k) = \frac{\partial \mathcal{L}}{\partial \beta_k}$ is known as the score function and the elements of the Fisher information matrix, $\mathcal{I}$, are given by

$$\mathcal{I}_{kl} = -E \left[\frac{\partial^2 \mathcal{L}}{\partial \beta_k \partial \beta_l} \right]. \tag{5.31}$$

In practice, a line search is required in order to ensure that at each iteration the

log likelihood is guaranteed to increase.

First derivatives of the log likelihood function

As shown in Petersen and Pedersen (2012), the derivative of the determinant, trace and inverse of a matrix, $\boldsymbol{A}$, are given by:

$$\frac{\partial}{\partial x}|\boldsymbol{A}| = |\boldsymbol{A}|\mathrm{tr}\left(\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\right), \quad \frac{\partial}{\partial x}\mathrm{tr}(\boldsymbol{A}) = \mathrm{tr}\left(\frac{\partial \boldsymbol{A}}{\partial x}\right), \quad \frac{\partial}{\partial x}\boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial x}\boldsymbol{A}^{-1}. \quad (5.32)$$

Using these identities, the derivative of the log likelihood function in Equation 5.29 with respect to an abstract parameter, β_k, is

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \beta_k} = &-\frac{1}{2}\sum_i^N c_{tot,i}\left[\boldsymbol{\Sigma}_{Z_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{Zi}}{\partial \beta_k} - \boldsymbol{\Sigma}_{Z_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{Zi}}{\partial \beta_k}\boldsymbol{\Sigma}_{Z_i}^{-1}\epsilon_i^2 + 2\boldsymbol{\Sigma}_{Z_i}^{-1}\epsilon_i\frac{\partial \epsilon_i}{\partial \beta_k}\right] \\ &-\frac{1}{2}\sum_i^N c_{tot,i}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_{XY_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{XYi}}{\partial \beta_k} - \boldsymbol{\Sigma}_{XY_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{XYi}}{\partial \beta_k}\boldsymbol{\Sigma}_{XY_i}^{-1}\left(\boldsymbol{S}_i + (\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi})(\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi})^T\right)\right)\right] \\ &-\frac{1}{2}\sum_i^N c_{tot,i}\left[2\,\mathrm{tr}\left(\boldsymbol{\Sigma}_{XY_i}^{-1}\left((\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi})\frac{\partial \boldsymbol{\mu}_{XY_i}{}^T}{\partial \beta_k}\right)\right)\right].\end{aligned} \quad (5.33)$$

The derivatives of the quantities $\boldsymbol{\Sigma}_{XY}$, $\boldsymbol{\mu}_{XY}$, $\boldsymbol{\Sigma}_Z$ and $\boldsymbol{\epsilon}$ with respect to the parameters, $\boldsymbol{\beta}$, are given in Appendix A.7 for the monochromatic wavelength model and Appendix A.8 for the Normal wavelength model.

Fisher information matrix

The Fisher information matrix is the negative of the expected value of the Hessian matrix of second derivatives. The expected values of the first and second moments of the distribution of the spots either side of the Ewald sphere surface are $E[\epsilon_i] = 0$ and $E\left[\epsilon_i^2\right] = \boldsymbol{\Sigma}_{Zi}$; the first and second moments of the general impact of the spot in the local reflection specific coordinate system are $E\left[(\bar{\boldsymbol{x}}_i - \boldsymbol{\mu}_{XYi})\right] = 0$ and $E[\boldsymbol{S}_i] = \boldsymbol{\Sigma}_{XYi}$. The elements of the Fisher information matrix only require the first derivatives and can be calculated as follows

$$
\begin{aligned}
\mathcal{I}_{kl} &= -E\left[\frac{\partial^2 \mathcal{L}}{\partial\beta_k \partial\beta_l}\right] \\
&= \frac{1}{2}\sum_i^N c_{tot,i}\left[\boldsymbol{\Sigma}_{Z_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{Zi}}{\partial\beta_k}\boldsymbol{\Sigma}_{Z_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{Zi}}{\partial\beta_l} + 2\boldsymbol{\Sigma}_{Z_i}^{-1}\frac{\partial \boldsymbol{\epsilon}_i}{\partial\beta_k}\frac{\partial \boldsymbol{\epsilon}_i}{\partial\beta_l}\right] \\
&+ \frac{1}{2}\sum_i^N c_{tot,i}\left[\mathrm{tr}\left(\boldsymbol{\Sigma}_{XY_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{XYi}}{\partial\beta_k}\boldsymbol{\Sigma}_{XY_i}^{-1}\frac{\partial \boldsymbol{\Sigma}_{XYi}}{\partial\beta_l} + 2\,\boldsymbol{\Sigma}_{XY_i}^{-1}\left(\frac{\partial \boldsymbol{\mu}_{XYi}}{\partial\beta_k}\frac{\partial \boldsymbol{\mu}_{XYi}}{\partial\beta_l}^T\right)\right)\right].
\end{aligned}
\tag{5.34}
$$

5.2.5 Integration

Once the profile model has been determined and the unit cell and orientation of the crystal have been refined, the reflections are then predicted on the image and subsequently integrated. The procedure is the same as described in Winter *et al.* (2018). For each of the predicted reflections, a shoebox is generated containing a set of pixels constituting the foreground region of the spot and a set of pixels constituting the background region of the spot. A background model is then estimated from the background pixels (Parkhurst *et al.*, 2016). This background model is then applied to the foreground pixels and the "summation" intensity is estimated as the total background subtracted pixel counts within the foreground region. Currently, no profile fitting is done; however, the profile model affects the spot prediction and the delineation of the foreground mask for in the reflection shoebox. The prediction and mask calculation are described below.

Prediction

Reflections expected to be observed on the image are predicted using the following procedure. First, a list of all possible Miller indices are generated out to a user specified resolution cut off. In the absence of a resolution cut off, the maximum resolution is determined from the maximum resolution at the corners of the detector. For each reflection, the distance from the central Ewald sphere is then calculated and if the reflection is within the χ^2 quantile with probability 0.9973 (corresponding to 3σ in the univariate case) then the reflection is predicted; otherwise, the reflection is not predicted, as shown in Figure 5.8. Larger reflection profiles in reciprocal space will result in more spots being predicted on the image.

Mask calculation

For each predicted reflection, a mask is calculated that separates pixels into foreground and background pixels. The foreground pixels are those pixels which, when mapped to the local reflection specific coordinate system, are within the χ^2 quantile with

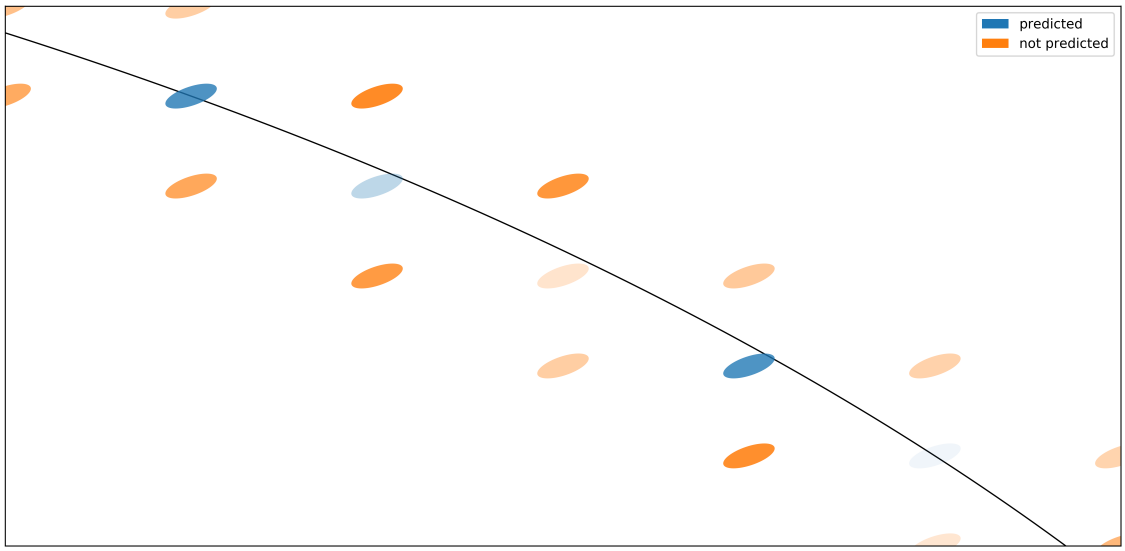


Figure 5.8: Spot prediction. Reflections close to the Ewald sphere (the solid black line) are predicted; those more distant from the Ewald sphere are not predicted. The distance is dependent on the variance of the reciprocal lattice point distribution. In the diagram, the shape of the RLP distribution is the same for each RLP; however, it should be noted that the intensities are different for each RLP, as indicated by the different colours of the reflections profiles.

probability 0.9973 (corresponding to 3σ in the univariate case). The shoebox is then expanded to contain a number of background pixels around the foreground region to be used in the background calculation. The shape of the foreground mask on the detector depends on the reflection profile model and the orientation of the detector.

5.3 Analysis

5.3.1 Experimental data

In order to evaluate the effect of the profile modelling algorithm on SSX data, three datasets were selected. An example of a spot at 3Å for each dataset is shown in Figure 5.9.

1. In order to provide a simple evaluation of the algorithm, 1000 images were simulated using the *simtbx* package within *cctbx* (Grosse-Kunstleve *et al.*, 2002), which provides a wrapper for the *nanobragg* program (Holton, 2018). Each image was simulated independently using identical parameters and a crystal with space group $P2_1$ and unit cell parameters $(39.5, 78.5, 48.0, 90, 97.8, 90)$. For each image, the crystal orientation was randomly assigned by uniformly sampling from misset angles between 0 and 2π. The benefit of using simulated data in the analysis is that the entire geometry of the experiment is known; therefore, the results of the profile model refinement procedure can be evaluated

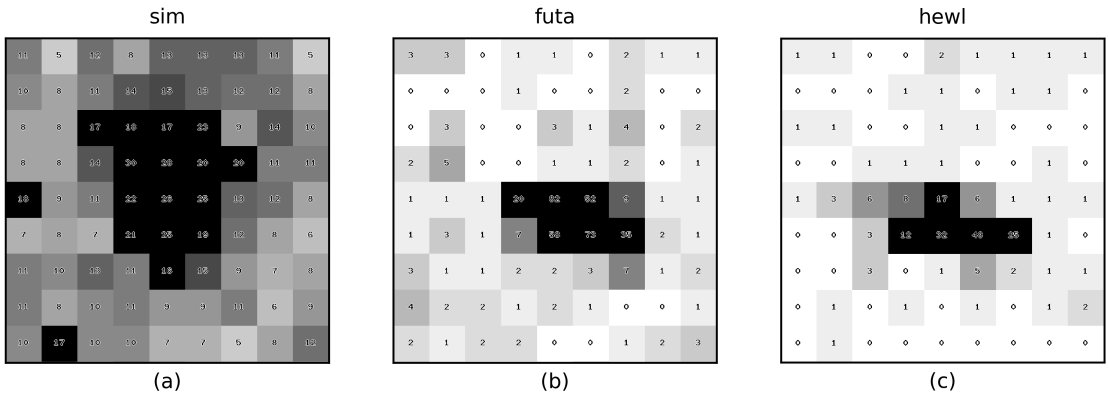


Figure 5.9: An example of a 3Å spot from the simulated dataset (a), FutA dataset (b) and HEWL dataset (c).

against the known experimental geometry.

2. FutA, an iron-binding protein (Polyviou *et al.*, 2018) collected on Diamond beamline I24. The dataset consists of 12,800 images with space group $P2_1$. The median unit cell parameters determined from refinement using *dials.potato* are $(39.54, 78.33, 48.16, 90, 97.79, 90)$. Data courtesy of Rachael Bolton and Ivo Tews, Southampton University.

3. A lead bound hen egg white Lysozyme (HEWL) dataset collected on Diamond beamline I24. The dataset consists of 17,828 images with space group $P4_32_12$. The median unit cell parameters determined from refinement using *dials.potato* are $(79.13, 79.13, 38.26, 90, 90, 90)$. Data courtesy of John Beale and Danny Axford, Diamond Light Source.

5.3.2 Data analysis

Each dataset was processed according to the following procedure:

1. Each image in the dataset was processed independently using *dials.stills_process* with the standard profile model algorithm. The standard profile model assumes a spherical RLP and spherical cap mosaicity model. For the simulated data, since the exact beam and detector geometry was known *a priori* from the simulation, these quantities were fixed in the refinement and were not allowed to vary. Thus, the geometry refinement was only able to vary the crystal unit cell and orientation parameters to fit the data.

2. For the FutA and HEWL data, the experimental models for all images were then combined and a joint refinement of all the experimental models was performed

(Waterman *et al.*, 2016). The beam and detector models were assumed to be the same for each image. This enables more accurate determination of the beam and detector geometry. In the case of the simulated data, this was not necessary since the beam and detector geometry were already known.

3. For the FutA and HEWL data, each image was subsequently re-processed independently using *dials.stills_process* a second time with the standard profile model algorithm. The beam and detector geometry were fixed to a reference beam and detector model derived from the result of the joint refinement. Again, this was not necessary for the simulated data.

4. The successfully processed images were then selected. Of the input images, a number of images failed during indexing; the number of indexing failures for each dataset can be see in Table 5.1. For the experimental data, the major cause of processing failure was that there were too few strong spots on the image to determine the crystal unit cell and orientation. This was a particular problem for the FutA data where the majority of images could not be processed for this reason.

5. The successfully processed images were then re-processed using the *dials.potato* program implementing the enhanced profile model algorithm described in this chapter. The detector and beam geometry were fixed to the reference geometry derived from the joint refinement; therefore, only the crystal unit cell, orientation and profile model parameters were refined.

6. Finally, the images successfully processed by the enhanced algorithm were selected. The *dials.potato* program applies more stringent requirements on the number of strong spots to perform the refinement than *dials.stills_process*. The input spots are first mapped to reciprocal space and assigned a fractional Miller index. If the distance in fractional Miller indices from the mapped position to the nearest integer Miller index exceeds 0.3 then the spot is removed and labelled as unindexed. A minimum of 10 indexed spots are required for the refinement, otherwise the program will terminate with an error. Analysis of the individual images indicated that the failures most often occurred when the input spots could not be indexed by the input model. The number of discarded images is shown in Table 5.1.

More details about the data processing can be seen in Appendix A.9.

Table 5.1: Data processing statistics. The rows show the number of diffraction images, the number of images successfully integrated by *dials.stills_process*, the number of images subsequently successfully integrated by *dials.potato*. Then the number of images that failed to be processed by *dials.stills_process* and the number of images that additionally failed to be processed by *dials.potato*. Then for the images which were successfully processed by *dials.stills_process* but failed to be process by *dials.potato*, the average number of strong spots observed on the image and the average number of spots that could be indexed on the image, average number selected for refinement and average number rejected by centroid outlier detection.

	Simulated	FutA	HEWL
N images	1,000	12,800	17,828
N success stills process	995	2,378	15,518
N success potato	995	2,323	15,074
N failure stills process	0	10,422	2,310
N failure potato	0	55	444
Mean N spots in failure	N/A	40.4	21.3
Mean N indexed in failure	N/A	14.9	16.2
Mean N selected in failure	N/A	7.5	7.1
Mean N outliers in failure	N/A	7.4	9.1

5.3.3 Analysis of crystal orientations

For the simulated data, the crystal unit cell and orientation are known for each image in advance; therefore, the unit cell and orientation determined by the algorithm can be compared directly with the true values. Given a crystal with a known orientation matrix, $\boldsymbol{U}_1$, and estimated orientation matrix, $\boldsymbol{U}_2$, the difference in orientation can be compared simply by computing the rotation matrix that transforms one orientation to the other. This rotation matrix is simply $\boldsymbol{R} = \boldsymbol{U}_1\boldsymbol{U}_2^T$; the rotation angle is $\theta = \cos^{-1}\left(\frac{\mathrm{tr}(\boldsymbol{R})-1}{2}\right)$. Obviously, where the crystal orientation is better determined, the angular difference between the true and estimated orientation matrices will be smaller. Since, for the FutA and HEWL data, the "true" orientation of each crystal is unknown, it is only possible to perform this analysis for the simulated data. Figure 5.10 shows the distribution of the angular differences between the true and estimated orientation matrices for all images in the simulated dataset for the data processed with the standard algorithm and the enhanced algorithm. It can be seen that the variance of the angular differences is much smaller for the data processed using the enhanced algorithm. This indicates that the orientations of the crystals are better determined using the enhanced algorithm.

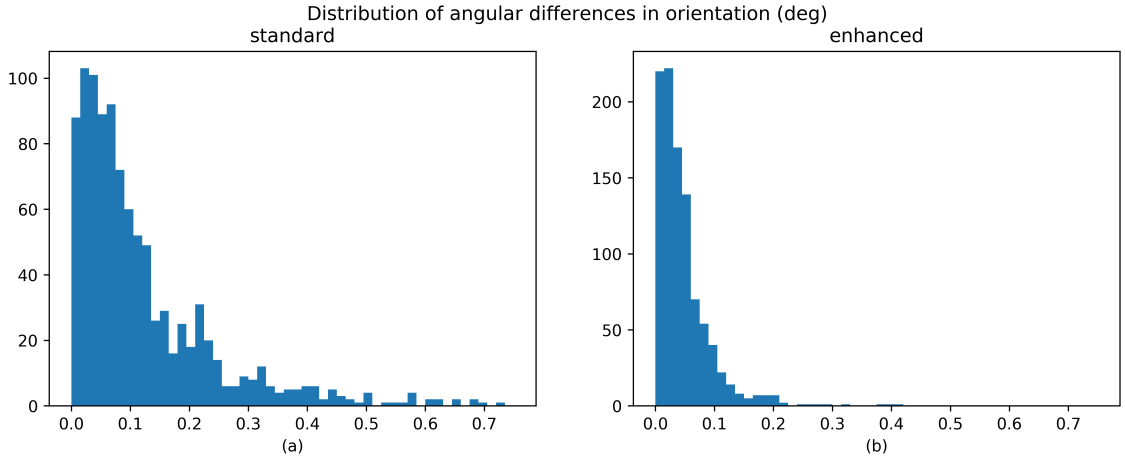


Figure 5.10: The angular difference between the known simulated crystal orientation and the refined orientation from (a) the standard algorithm and (b) the enhanced algorithm.

5.3.4 Analysis of crystal unit cell parameters

As before, for the simulated data, the true unit cell parameters are known and are the same for each simulated image. For the FutA and HEWL data, the true unit cell parameters are unknown and will indeed be different for each image. However, the distribution of the unit cell parameters can be analysed; since each image is refined independently, a tighter distribution of unit cell parameters may indicate that the unit cell parameters are better determined. The mean unit cell parameters can be used as a reference for comparing the unit cell parameters estimated by the refinement algorithms; however, this can be affected by outliers, so the median unit cell parameters are used instead.

Instead of comparing the unit cell parameters directly, it is instead convenient to compute the difference between the estimated reciprocal space orthogonalisation matrix, $\boldsymbol{B}_2$ and a reference, $\boldsymbol{B}_1$, which may either be derived from the known true unit cell parameters or the median unit cell parameters. The difference between these matrices can be quantified using the distortion index (Thompson *et al.*, 2018). This is a measure of the fractional distortion between the two matrices; a smaller number indicates a smaller distortion. Given the metrical matrices, $\boldsymbol{G}_1 = \boldsymbol{B}_1^T\boldsymbol{B}_1$ and $\boldsymbol{G}_2 = \boldsymbol{B}_2^T\boldsymbol{B}_2$, the distortion index, d, can be computed as follows:

$$\begin{aligned} \boldsymbol{D} &= \boldsymbol{G}_1\boldsymbol{G}_2^{-1}, \\ d &= \frac{1}{2}\left\|\boldsymbol{D} - \mathbb{1}\right\|_1 = \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\left|\boldsymbol{D}_{ij} - \mathbb{1}_{ij}\right|. \end{aligned} \tag{5.35}$$

Figure 5.11 shows the distortion index between the estimated reciprocal space orthogonalisation matrices and a reference matrix for each dataset. In each case, it

can be seen that the distortion index is significantly lower for the estimates provided by the enhanced algorithm than for the estimates provided by the standard algorithm. However, the improvement is larger for the simulated data than for the FutA data and HEWL data. For the simulated data, the enhanced algorithm results in a 70% reduction in the standard deviation of the distortion index over the standard algorithm (from 0.0101 to 0.0030). For the FutA data, the reduction is 27% (from 0.0049 to 0.0036) and for the HEWL data, the reduction is 7% (from 0.0027 to 0.0026). Since, in the case of the FutA and HEWL data, the median unit cell parameters are used rather than the true unit cell parameters for each crystal (which are unknown), it may be that the variance in the estimated unit cell parameters is dominated by the variance in the "true" unit cell parameters or the trade-off between the unit cell and orientation parameters. Therefore, the distortion index between the estimated unit cell and the median unit cell is dominated by the intrinsic difference between the unknown "true" unit cell and the median unit cell.

The spread in the estimates of the individual unit cell parameters for each dataset can also be seen in Figure 5.12. Here it can be seen that, for the simulated data and the FutA data, the variance in the unit cell parameter estimates is lower using the enhanced algorithm than using the standard algorithm. In particular, the spread in the estimated unit cell angles is much lower in both cases. For the HEWL data, the spread in the unit cell parameters does not differ much between the two algorithms; this may be due to the fact that the higher symmetry means that fewer parameters need to be determined in the refinement, meaning the remaining parameters may be better determined.

5.3.5 Analysis of positional residuals

The accuracy of the algorithm in predicting the positions of spots on the image was assessed by comparing the observed positions of the spots on the images with their predicted positions. The observed position of a diffraction spot is determined by calculating the intensity weighted average "centroid" of the pixels contributing to the spot. The observed position is then dependent on the pixels which are used to compute the centroid. In order to avoid biased centroid estimates, the pixels used to determine the spot centroid must not be biased towards the predictions from either algorithm. Additionally, the centroids of weak spots are poorly determined so only strong spots were used in the analysis.

For the simulated data, it was possible to extract very accurate "true" observed positions of the diffraction spots; the simulator produces both a diffraction image with various sources of noise and an ideal image without any noise. To compute the observed centroids, the noise-free images were used. In the noise-free images, the

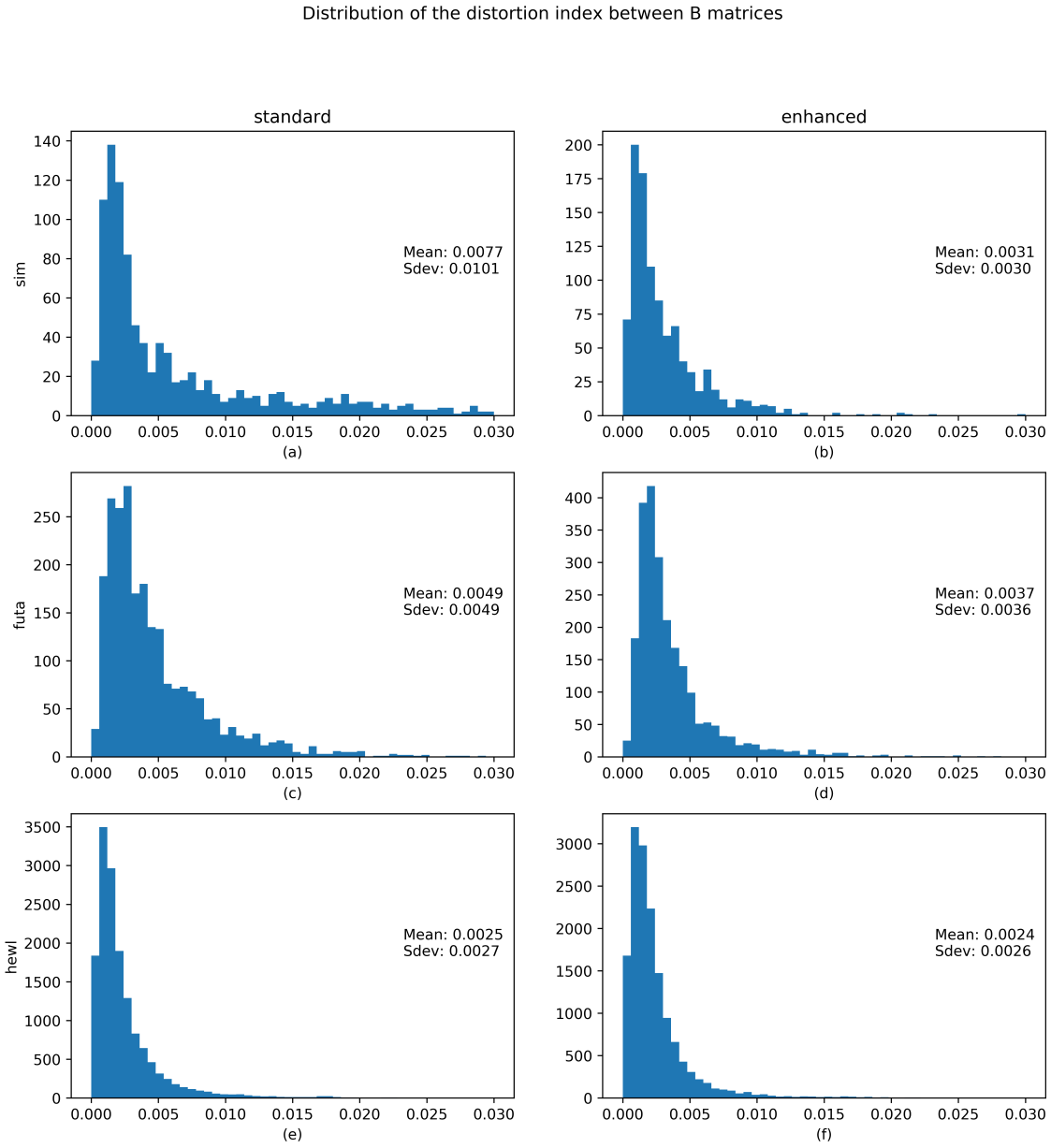


Figure 5.11: The distortion index (Thompson *et al.*, 2018) between the known or median $\boldsymbol{B}$ matrix and the refined $\boldsymbol{B}$ matrix for the standard algorithm (left) and the enhanced algorithm (right) for the simulated data (top), FutA data (middle) and HEWL data (bottom).

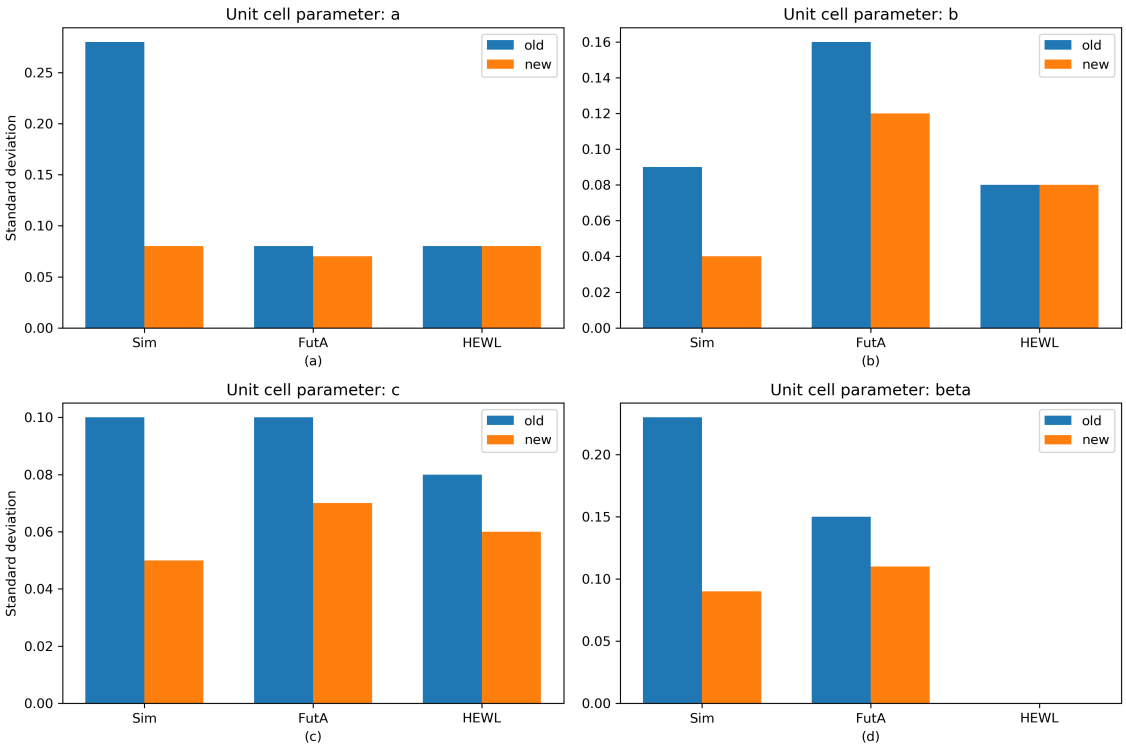


Figure 5.12: The standard deviation of estimated unit cell parameters from the standard algorithm (old) and the enhanced algorithm (new). For the simulated data and the FutA data, the space group is $P2_1$ and the alpha and gamma unit cell angles are exactly equal to 90 degrees. For the HEWL data, the space group is $P4_32_12$ and all the unit cell angles are 90 degrees.

background pixels are all zero so the pixels contributing to the individual spots can be segmented very easily. The observed position of each spot is then the intensity weighted average of the non-zero pixels. For the FutA and HEWL data, the observed spot positions must be estimated from the noisy data. This is done by selecting pixels in the vicinity of a spot and computing the background subtracted intensity weighted average pixel position. In order to avoid bias in the centroid positions, the following procedure is performed.

- The set of strong spots found during the initial spot finding is selected. An initial estimate of the observed spot position is calculated from the "strong" pixels selected by the spot finding algorithm.
- For each strong spot, a shoebox of size 21x21 pixels, centred on the observed centre of mass, is computed. A mask containing all pixels whose centres lie within 6 pixels of the observed centre of mass are assigned as foreground pixels; the remaining pixels are assigned as background pixels.
- The background within the foreground region of the shoebox is then estimated from the set of background pixels.
- An updated observed position is determined by computing the background subtracted intensity weighted average centroid of the foreground pixels. The shoebox is then re-centred on the new observed position and the procedure is iterated until the observed position of the spot changes by less than some small value. This is typically reached after a few iterations.

The residuals between the observed and predicted spot positions in the X and Y detector positions were then calculated. Additionally, the radial and transverse components of the residuals relative to the direct beam position on the detector were calculated. Since a different set of reflections may be predicted for the standard and enhanced algorithms, the set of common reflections was used in this analysis.

The X, Y, radial and transverse RMSDs were then calculated for each image for both the standard and enhanced algorithm. The difference between the RMSDs for the standard and enhanced algorithms were then calculated to determine if there was any improvement in the prediction of the spot positions. Figure 5.13 shows the distribution of differences in the different components of the RMSDs for each dataset. For each dataset, the average RMSD for the enhanced algorithm is lower than the average RMSD for the standard algorithm. This is the case for the X, Y, radial and transverse RMSDs. For the simulated data, the improvement is roughly the same for each component; however, for the FutA and HEWL data, the RMSDs in the X direction appear to have improved more than the RMSDs in the Y direction. The

reason for this is unclear but may be related to the synchrotron X-ray beam profile on Diamond beamline I24 which differs in the horizontal and vertical directions (which are roughly aligned with the X and Y directions on the detector). The RMSD in the radial direction also shows a greater improvement than the RMSD in the transverse direction.

5.3.6 Analysis of predictions

Each still diffraction image represents a slice through reciprocal space. The set of reflections that are predicted to be observed on the image is model dependent; given a RLP that is a distance from the Ewald sphere, some criteria for determining that the spot is close enough to be observed is imposed. Therefore, different profile models produce different sets of predicted reflections. Ideally, all the spots that are observed on the image should be predicted by the algorithm. Observed diffraction spots may be unpredicted for a number of reasons, for example, the diffraction image may contain contributions from multiple crystals which haven't been identified or there may be pathologies in the crystal that are difficult to model; however, since under-prediction leads to a loss of information, for the purposes of integration, all observed diffraction spots should be integrated with minimal over-prediction. Excess reflections that are zero or have very low partiality can be handled during scaling or post-refinement.

Figure 5.14 shows the fraction of observed strong spots that have been predicted by the standard and enhanced algorithms for each dataset as a function of the total number of observed strong spots. It can be seen that, in each case, the enhanced algorithm results in the successful prediction of a larger fraction of observed strong spots than the standard algorithm. Indeed, for the FutA and HEWL datasets, the standard algorithm drastically under-predicts the number of spots observed on the diffraction images, with the enhanced algorithm often predicting more than twice the number of observed strong spots. For the simulated data, the number of unpredicted spots is lower; however, the enhanced algorithm still performs better.

Figure 5.15 shows the fraction of strong spots predicted in resolution bins. Across the entire resolution range, the enhanced algorithm predicts more of the observed strong spots than the standard algorithm. For the simulated data, the enhanced algorithm essentially predicts all the observed spots, with the standard algorithm slightly under-predicting. For the FutA data, the enhanced algorithm predicts more than 85% of the observed strong spots across the entire resolution range. At low resolution, it slightly under-predicts the number of observed strong spots; however at high resolution, more than 90% of observed strong spots are predicted. For the HEWL data, the enhanced algorithm predicts more than 90% of observed strong spots across

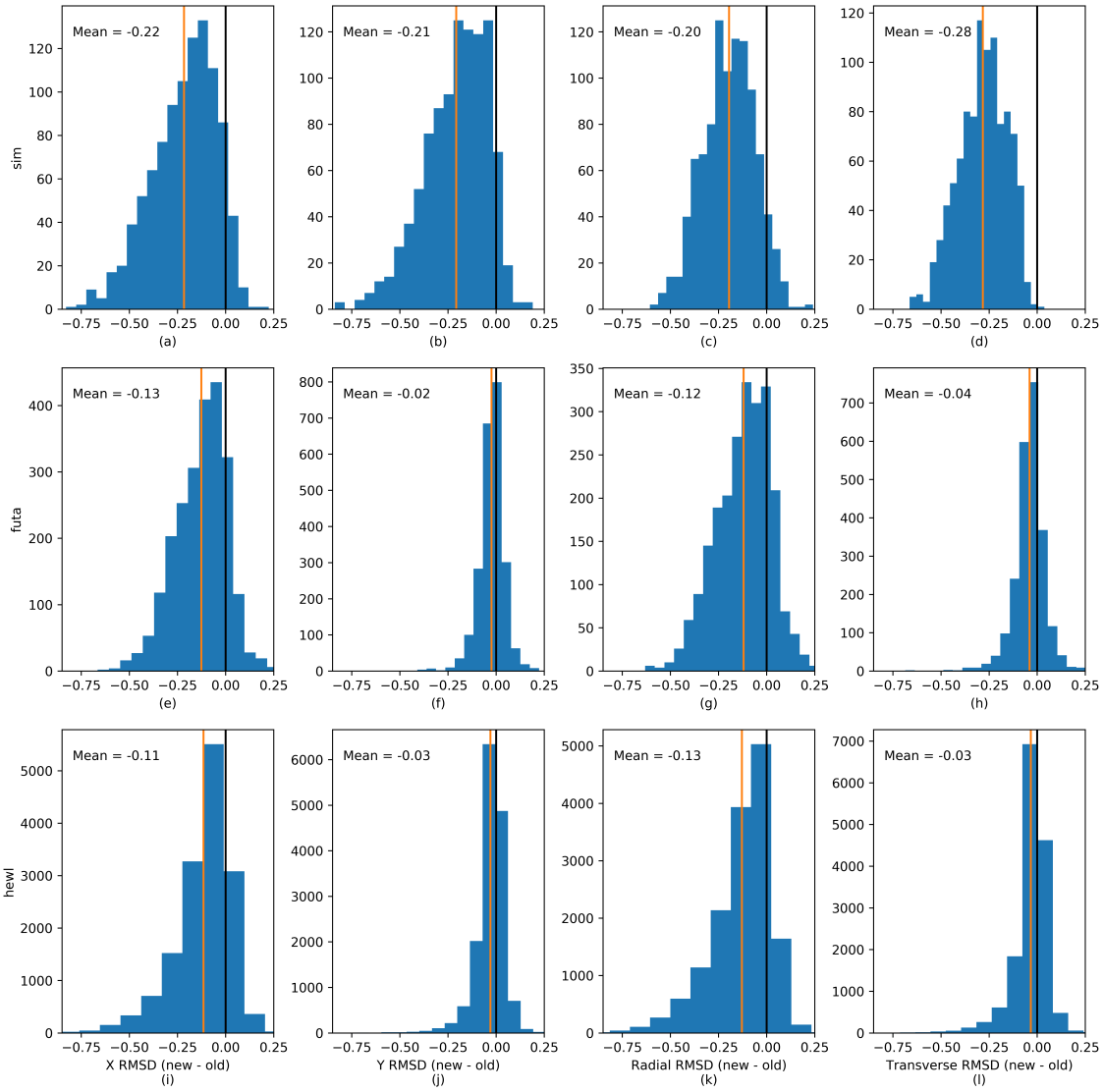


Figure 5.13: The distribution of the difference in X (column 1), Y (column 2), radial (column 3) and transverse (column 4) RMSDs for the simulated data (top), FutA data (middle) and HEWL data (bottom). Here, old refers to the standard algorithm and new refers to the enhanced algorithm. The black line indicates zero difference in the RMSD and the red line indicates the mean difference in the RMSD. In each case, the mean difference in the RMSD is negative indicating that the standard RMSD is larger than the enhanced RMSD.

the entire resolution range. In contrast, for both the FutA and HEWL data, the standard algorithm under-predicts the number of observed strong spots; only between 65% and 80% of observed strong spots are predicted across the whole resolution range. An example image for each dataset is shown in Figure 5.16 showing the strong and weak spots predicted by the enhanced algorithm and the spots predicted by both algorithms. This provides an illustration of the extent of the improvement to the set of predicted spots by the enhanced algorithm over the standard algorithm.

5.4 Conclusion

In the processing of still X-ray diffraction data, the positions of the reflections on the detector depends on the form of the profile model used for the reflections. The use of a Multivariate Normal distribution to describe the shape of the reciprocal lattice points combined with a Normal distribution model for the spread in wavelengths has been presented. The algorithm was evaluated through its use on three datasets: a simulated dataset generated by the *nanobragg* simulator, and two still image datasets collected on Diamond beamline I24 with a fixed target setup. It has been shown that the use of this model can result in an improvement in the determination of the crystal unit cells and orientations. Use of the profile model also results in better prediction of reflection positions on the detector and better prediction of the set of reflections actually present. The method is implemented within *DIALS* and can currently be used *via* a stand-alone program *dials.potato* that is run after *dials.stills_process* as detailed in Appendix A.9.

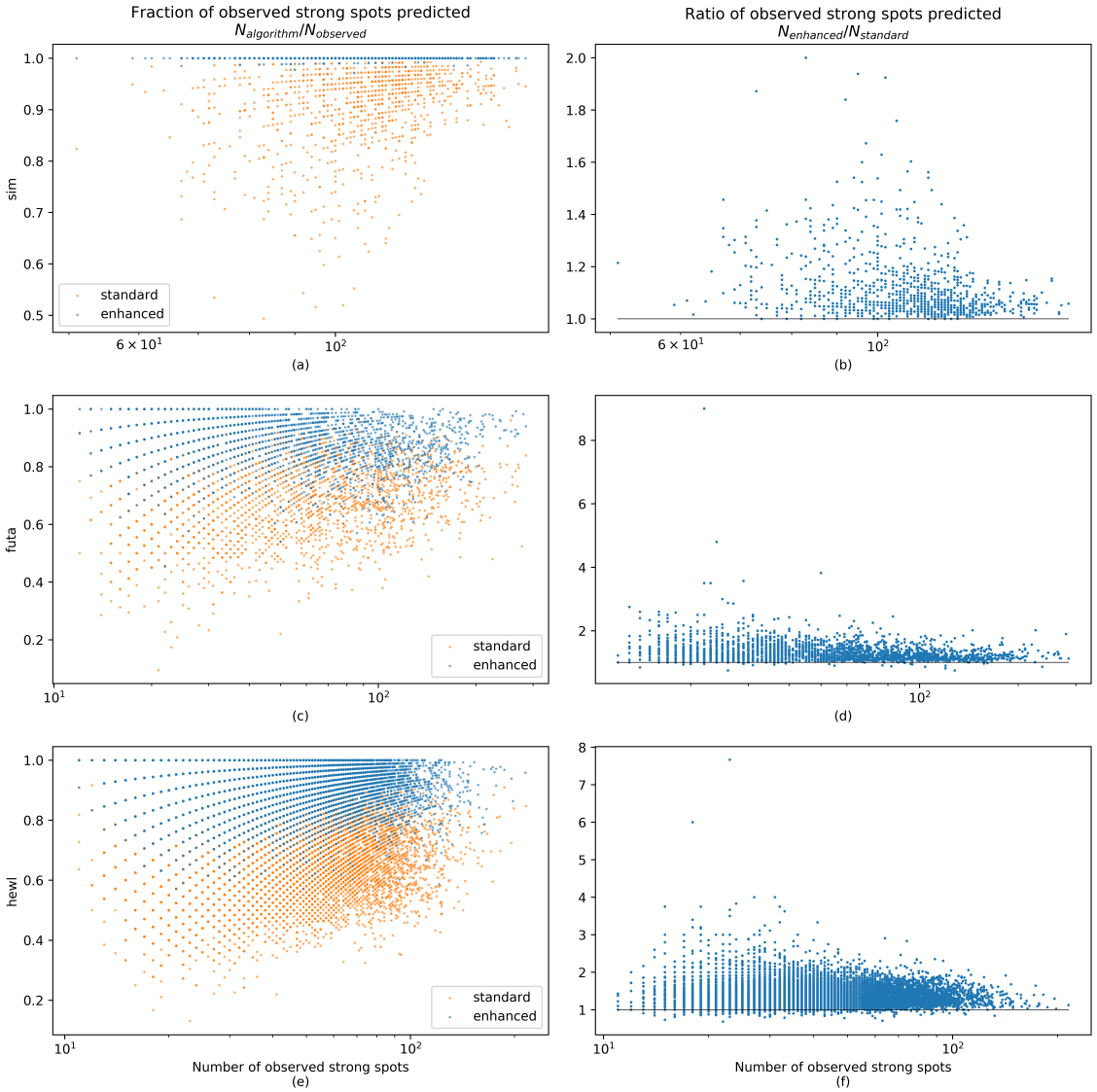


Figure 5.14: A comparison of the predictions from the standard and enhanced algorithms. The fraction of strong spots observed on the images that were predicted by each of the algorithms for (a) the simulated data, (c) the FutA data, and (e) the HEWL data. The ratio of the number of number of strong spots predicted for the enhanced algorithm and the number of strong spots predicted by the standard algorithm for (b) the simulated data, (d) the FutA data, and (f) the HEWL data.

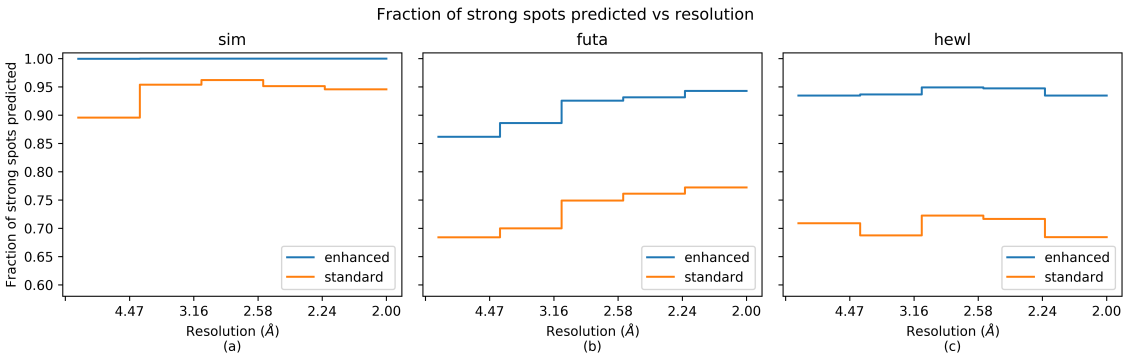


Figure 5.15: The fraction of strong spots predicted by both the algorithms in resolution bins for (a) the simulated data, (b) the FutA data, and (c) the HEWL data.

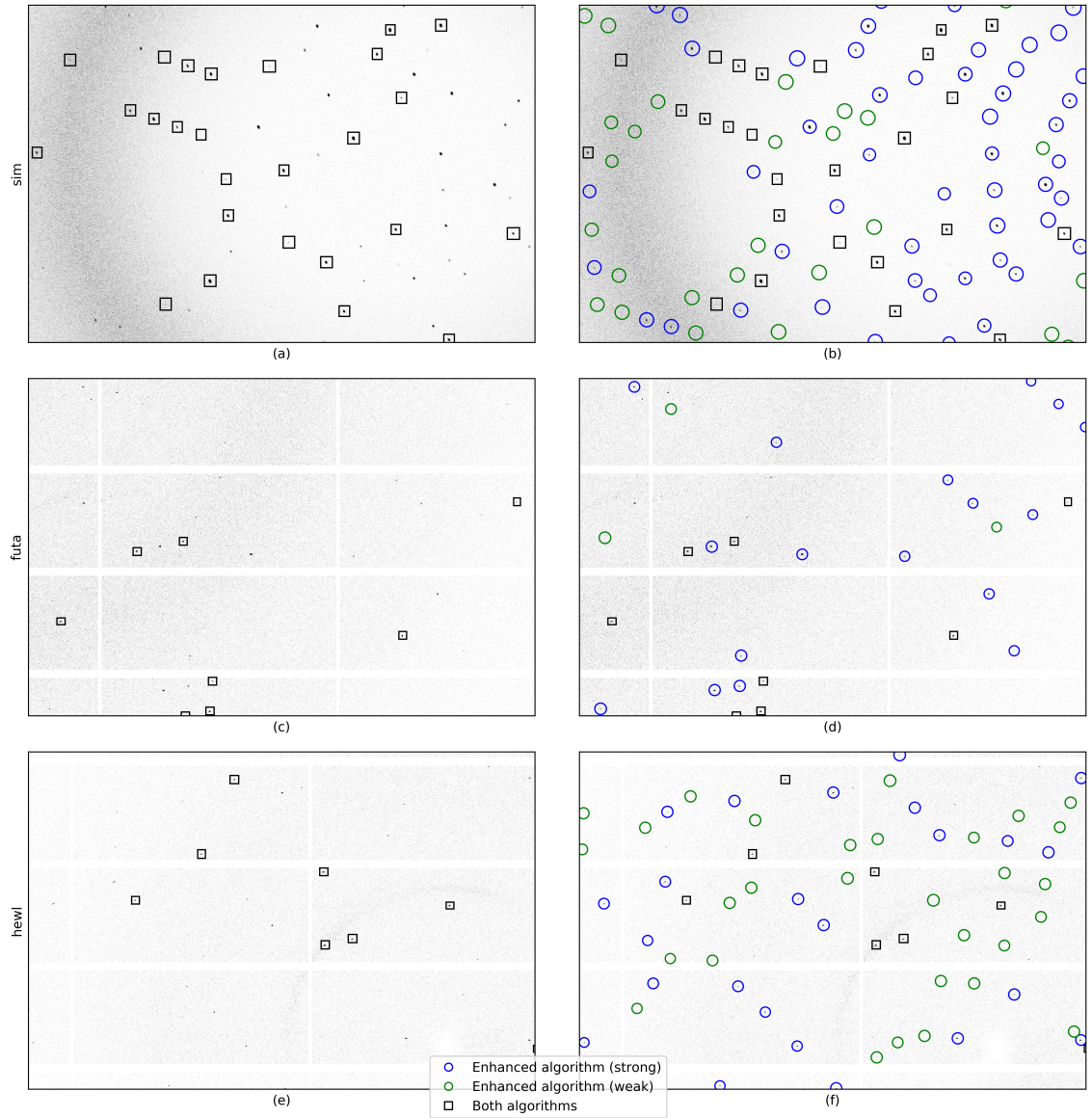


Figure 5.16: An example from each dataset for the spots predicted by the standard and enhanced algorithms.

Chapter 6

Discussion

6.1 Conclusions

The aim of the project was to develop statistically robust algorithms for the integration and analysis of X-ray diffraction data in order to address new developments in data collection and detector technology resulting from new and emerging challenges in macromolecular crystallography. These new trends are driven in part by the constantly improving ability for structural biologists to collect data from ever smaller crystals resulting in weaker, noisier, but still measurable, diffraction.

New photon counting pixel array detector technology has resulted in a revolution in data collection for X-ray crystallography. These detectors have very fast readout times and essentially zero readout noise allowing complete, finely-sliced datasets to be collected within seconds. In these detectors, the pixels operate essentially independently resulting in a very small point spread function; additionally, the direct photon counting nature of the detectors means that even a single photon can be accurately measured. However, these properties mean that integration algorithms in data processing programs need to be modified in order to correctly handle the data generated by these new detectors.

Smaller crystals result in an increased prevalence of radiation damage which means that it may not be possible to collect a full rotation from a single crystal. The use of micro crystals then requires data from multiple crystals in order to create a complete dataset. In the extreme case, this results in each single crystal producing a single diffraction image. This mode of data collection, known as serial crystallography, is becoming more popular at synchrotrons and provides another challenge to data processing programs.

6.1.1 Robust background modelling using Generalised Linear Models

In the integration of the reflection intensities, pixel outliers in the neighbourhood of the Bragg peak are assigned as either foreground pixels containing signal or background pixels. The background under the reflection peak is then estimated by applying a model derived from the set of nearby background pixels. In order to provide an accurate estimate of the reflection intensity, it is, therefore, necessary to ensure that the estimate of the reflection background is also accurate; this is complicated by the presence of pixels outliers, such as zingers, ice rings and unmodelled intensity from adjacent reflection, in the reflection background. If these pixels are used within the background calculation, then the background estimate will be positively biased resulting in an underestimation of the reflection intensity. Therefore, they need to be properly handled by the background procedure.

The outlier handling methods described in the literature of integration programs explicitly or implicitly assume a Normal distribution. However, for very weak data collected on pixel array detectors, where the average number of background counts may be less than a single count per pixel, this assumption is not appropriate: only when the number of counts is large, is the Normal distribution a good approximation to the Poisson distribution. Use of these methods, causes an underestimation of the reflection background and an overestimation of the reflection intensities. In extreme cases, the background for every reflection may be incorrectly estimated as zero. This positive bias in the reflection intensities causes data processing statistics to look superficially better since adding a positive constant to all reflections makes symmetry equivalent reflections appear more similar; however, it can also give the false impression that the data are twinned, especially at high resolution where the data are weaker.

An algorithm was developed using a robust Generalised Linear Model (GLM) framework to estimate the reflection background in the presence of pixel outliers. Within the GLM framework, the data are explicitly assumed to be Poisson distributed, allowing the data to be handled in a principled way. Use of the algorithm results in a significant improvement in the estimated background and removes the positive bias in the integrated reflection intensities. The algorithm is very robust and enables better determination of weak reflection intensities at high resolution, allowing more information to be extracted from weak datasets. This can potentially have a significant impact on the refined electron density. The algorithm is the default background algorithm in *DIALS*.

The algorithm is designed for use with Poisson distributed data. This is appropriate for photon counting pixel array detectors which are now the norm at

synchrotron facilities; however, other detectors, such as CCDs, may have different properties. Integrating detectors convert a total analogue charge into a number of counts; during this process, a pedestal may be subtracted and a multiplicative gain applied. Problems can occur if the detector is poorly calibrated; if the pedestal subtracted is larger than the number of counts in a pixel then the pedestal subtracted pixel will have a negative value. In a strictly Poisson framework, negative pixel values are not permissible. In some extreme cases, the detector may be so poorly calibrated that a significant number of pixels are negative and the resulting background estimate for some reflections may itself be negative; this means that subtracting the background actually increases the total signal pixel counts! Whilst it could be argued that, in these cases, the detector should be better calibrated, users expect that the data processing software should still provide a result. Therefore, if the data are pathological and contain negative counts, the background estimate is done *via* a fallback to a method assuming a Normal distribution. Another limitation is that, currently, the GLM estimator for a non-constant background is computationally expensive; a faster implementation is needed.

6.1.2 Background modelling in the presence of ice-rings

The background under the reflection peak is typically modelled as a constant or a plane. Since the X-ray background tends to vary slowly on the scale of a single reflection on the detector, these models are generally appropriate; however, there are occasions when these simple models are no longer adequate. A common pathology in X-ray diffraction datasets is the presence of water ice rings. These ice rings result in a sharp variation in the background in the radial direction away from the beam centre; a simple constant or plane background model is then no longer sufficient to describe the variation of the background counts underneath the reflection peak. If a simple background model is used then the background will tend to be over-estimated for reflections lying on the edge of the ice ring, since pixels at the ice ring peak will contribute to the background model estimation. Conversely, the background will tend to be under-estimated for reflections lying on the peak of the ice ring, since only pixels at the edge of the ice ring will contribute to the background model estimation. This will result in systematic errors in the integrated reflection intensities; this can be readily seen in the scatter plots of intensity as a function of resolution produced by *AUSPEX* (Thorn *et al.*, 2017). One approach to deal with reflections lying on the ice rings is to exclude all reflections within a resolution range covering the ice ring; however, this inevitably leads to a loss of information which may cause a lack of completeness, problems in refinement and distortion in the electron density maps. A better solution is to correctly model the shape of the ice ring during the background

model determination such that the ice ring contribution can be correctly subtracted during integration.

A global background modelling algorithm for X-ray diffraction data was implemented. The algorithm operates by computing an average value for each background pixel. The average background is then processed and smoothed along lines of constant resolution to produce a background model for use in integration. The background model is then fit locally to model the background of each reflection using a robust estimator to reduce the deleterious impact of pixel outliers in the reflection background. The use of the global background model algorithm reduces the negative impact of ice rings on the estimation of the reflection intensities: the systematic error in the intensities is much reduced or eliminated. A counter intuitive result is that correctly handling the ice rings in the reflection background can result in worse overall merging statistics. This effect is due to the fact that adding a positive constant to all symmetry equivalent reflections causes those intensities to appear to be more similar; the R merge, which is a measure of similarity between symmetry equivalent reflections then tends to improve. However, correct handling of the ice ring background is beneficial for refinement and can result in an improvement to the R_{work} and R_{free} and better quality electron density maps.

Currently, the global background modelling algorithm only supports a static model of the X-ray background. Since the X-ray background is expected to vary over the course of a scan, an obvious extension to the method would be to implement a scan-varying background model which could take into account these variations. The global background modelling algorithm is more computationally intensive than the standard local background modelling approach; indeed it requires a separate pass over the image data in order to first construct the global background model. The background modelling is also currently run as a separate manual processing step and is therefore not generally included in automatic data processing pipelines by default. However, it could be easily integrated by including a check for the presence of ice rings before optionally processing the data using the global background model algorithm.

6.1.3 Profile modelling for serial synchrotron X-ray diffraction data

Due to the increasing use of micro crystals, for which radiation damage is a significant issue, multi crystal data collection strategies have become more popular. In the extreme case, this can mean a crystal per diffraction image. At XFELs, where all the X-rays are delivered in a femtosecond pulse, it is not possible to rotate the crystal in the beam and the data are a set of still diffraction images covering a slice

through reciprocal space. This mode of data collection is also becoming more popular at synchrotrons, where it is called serial synchrotron crystallography (SSX). Serial crystallography at synchrotrons can refer to both small rotations or still images. Whilst it may be preferable to perform a small rotation, since a larger, better defined, region of reciprocal space will be covered, there are some occasions in which it is preferable to collect still images. For example, collecting still images is simpler from a hardware perspective and the experiment can be easily ported to an XFEL beamline, allowing the same experiment to be performed at both a synchrotron and an XFEL.

However, the processing of still image diffraction data presents a set of unique challenges that requires the development of new algorithms. If the reflection profiles in reciprocal space are non spherical, and the wavelength distribution is monochromatic, then the centre of mass of the spot will be given by the conditional distribution of the reciprocal lattice point on the Ewald sphere. This depends on the exact form of the reflection profile and, depending on the orientation of the reflection in reciprocal space, this will result in a different centroid being recorded on the detector. This then implies that centroid based refinement of experimental geometry and crystal unit cell and orientation also requires profile information. There are further complications if there is a spread of wavelengths. A larger spread of wavelengths will result in larger spots and will also change the centre of mass of the spot as recorded on the detector. Additionally, the set of spots predicted to be observed on the diffraction image is dependent on the reflection profile model; a larger profile results in a greater number of reflections being predicted to have some fraction of their total intensity recorded on the image. Therefore, a reflection profile model needs to be used in order to perform the refinement and reflection prediction.

A model was developed utilising an anisotropic 3D multivariate Normal distribution to model the shape of the reflection profile in reciprocal space along with a Normal distribution to model the spread of X-ray wavelengths. The parameters of the model are estimated *via* a maximum likelihood algorithm which takes into account the centre of mass of the spot as well as the extent of the spot on the detector (*i.e.* its general impact) and its distance from the mean Ewald sphere. Use of the algorithm results in reduced spread of unit cell parameters, indicating that the unit cells may be better determined relative to the standard profile model algorithm. It also results in improved predictions for the reflection centroids. Finally, it results in better prediction of the set of observed spots, relative to the standard profile model algorithm which tends to under-predict the set of spots visible on the detector. A limitation of the algorithm is that it doesn't take into account scaling and partiality (*i.e.* post refinement is not performed). Once fully implemented, this may result in better determination of the crystal unit cell and orientation.

6.2 Future work

The algorithms presented here all have scope for future development and improvement. The GLM background modelling could be improved by extending efficient implementations to non-constant models. Additionally, some detectors have a pedestal that is subtracted from the raw pixel counts resulting in negative pixel counts. Better analysis of the distribution of these pixel counts could yield better statistical methods for these data. This would require the uncertainty in the pedestal subtraction to be estimated. Another challenge that needs to be addressed is the development of new integrating detectors implementing dynamic gain variation, where the gain is varied according to the amount of charge deposited on the pixel (Mozzanica *et al.*, 2018).

The global background modelling algorithm could be extended by implementing a scan varying global background model which describes the variation in the background across both the detector and the scan. The software could also be made more user friendly by implementing better integration of the code into automatic data processing pipelines and graphical user interfaces.

The reflection profile modelling algorithm for synchrotron still images might be improved by better utilisation of reflection intensity information. However, post refinement for still image diffraction data is known to be problematic, particularly in the calculation of reflection partialities. This is especially the case for data collected at synchrotrons with a monochromatic beam where the notion of partiality is not well defined. More generally, the processing of stills image data within the *DIALS* framework has scope for improvement. This entails the development of better algorithms but also requires more user friendly tools and better automation; indeed, this is a frequent request from beamline scientists and users. In the context of SSX in general and the *DIALS* project in particular, there is a real need for better methods for scaling and merging of still X-ray diffraction data. One approach being investigated is to avoid the direct estimation of reflection partiality from the experimental geometry by fitting a functional form directly to the observed partiality.

The reflection profile modelling algorithm for stills currently assumes a unimodal spot shape; however, it is common to have split crystals that result in split diffraction spots on the image. The algorithm could be extended in order to properly model these split spots. This may also have application for the processing of rotation data; particularly during profile fitting, where proper modelling of the split spots may improve the intensity estimation. Currently, implementing a profile fitting algorithm for still images data is problematic. For rotation data, the reference profiles are typically generated *via* empirically averaging the profiles of strong reflections. This generally requires knowledge of the fully recorded profile of a large number of strong diffraction spots; however, in stills diffraction data, there is only a single slice through

reciprocal space and no spots are fully recorded. Furthermore, there are far fewer strong reflections for use in generating the reference profiles than there are for rotation data. In addition to this, since the crystal unit cell and orientation parameters are typically less well determined for still data than for rotation data, the resulting uncertainty in the spot positions will lead to larger errors in the reference profiles and profile fitted intensities. So profile fitting for still images presents problems in both building the reference profiles from the partial reflections and then fitting them to partial reflections. An approach to generate reference profiles might be to use reference profiles generated from multiple still images to perform profile fitting. If this could be done then it may improve the estimated intensities from the stills data. However, the validity and consistency of using profiles from many crystals would need to be ascertained.

With each new generation of detectors, there is an increase in the rate of data collection. There is then a corresponding demand from beamline scientists and users for data processing software to keep up with the speed of data collection in order to provide live feedback that can help to inform and guide the experiment. Moore's law no longer holds (Waldrop, 2016): the speed of serial execution is not increasing fast enough for data processing to keep up with the output of new detectors. Therefore, data processing programs need more efficient algorithms and better exploitation of parallelism in to order to achieve significant performance gains. The challenge of high data rate crystallography is only likely to increase.

Data processing in macromolecular crystallography is a mature field; however, some challenges still remain. These challenges are often driven by user requirements, such as extracting as much signal as possible from tiny crystals. There are also areas, as well as those already discussed where active research may still produce some benefit. For example, ray tracing approaches, if made efficient and robust, may offer a substantial improvement in terms of the accuracy and utility of profile fitting. To this end, whole image modelling, including modelling of the background could enable both better background determination and intensity estimation within a single consistent framework. This ray tracing scheme would require improvements in optimisation methods such as stochastic, Bayesian, modelling approaches. However, it might be beneficial in cases where there isn't enough data to generate empirical reference profiles for profile fitting, such as in still X-ray diffraction images and small rotations.

6.3 Impact, deployment and use of the software

It has been an aim of this project to deliver production quality software for both individual users and for deployment on synchrotron beamlines within automated data

processing pipelines. To that end, the algorithms and software have been designed to be generally applicable as opposed to merely targeting a specific experiment, diffraction dataset or beamline. Within the *DIALS* project as a whole, a significant amount of effort was put into ensuring that the software works and is available across multiple computing platforms: the software can be run on Linux, Windows and Mac operating systems. This entailed a large amount of testing and debugging, and the production of a significant amount of user and developer documentation. It is good practice to do these things even if the software is only intended to be used by a single person or within a single lab: scientific results derived from the output of computer programs requires those computer programs to be robust, reliable and well tested. However, these practices are additionally important when the software is intended to be used by the wider community.

The *DIALS* framework has been developed as part of a collaboration with contributions from many software developers. As one of the first people involved in the project, I was responsible for implementing most of the low level core functionality of the *DIALS* framework on which many other developments have been built. Considerable time and effort was put into ensuring that these components were written to be robust and reliable with sensible user facing APIs for other developers to use. Finally, the *DIALS* framework contains a suite of data processing programs which are all required to behave consistently and seamlessly exchange data with one another. I designed the command line interfaces and implemented the data structures and data file serialisation to address these requirements.

The *DIALS* software itself exists as a set of command line programs; however, it is also incorporated into automated data processing pipeline software such as *xia2*, through which it is available to beamline scientists and users at Diamond Light Source and other synchrotrons. As such, the software is used to process essentially every dataset collected on every Diamond MX beamline. Given the increasing importance of automation, particularly in the context of remote data collection which is becoming more common, beamline users are increasingly likely to rely on automated pipelines to process their data. This is particularly the case given the ever increasing size of X-ray diffraction datasets produced by new generations of detectors. It is reaching the point where the storage and transfer of raw diffraction images makes it impractical for users to process them in the lab: data will increasingly be processed *in situ* at the synchrotron.

When the *DIALS* project was started, the only supported integration package within the *CCP4* suite was *MOSFLM*; at the time of writing, *DIALS* is now the main integration program within the *CCP4* suite; as such, *DIALS* is distributed and used throughout the world by *CCP4* users. Additionally, starting in 2015, tutorials on how to use *DIALS* have been given at every *CCP4* structure solution workshop

and *DIALS* has been used by workshop attendees to solve various difficult data processing problems. A common request from users, particularly at data processing workshops, has been that a graphical user interface (GUI) would be desirable to enable users who are unfamiliar with the command line to more easily process their data. In order to achieve this, the DIALS graphical user interface (DUI) is being developed by Luis Fuentes Montero (Fuentes-Montero, 2019).

At the time of writing, *DIALS* has been used to solve 241 structures deposited in the PDB; since many structure depositions only state *xia2* as the data processing program (which may mean either *DIALS* or *XDS*), it is difficult to determine the exact number. However, some analysis has revealed that more than 1000 further structures are likely to have be processed using *DIALS* (Graeme Winter, private communication). The software is open source, distributed under a BSD licence, and freely available for anyone and everyone to use. The source code is available online and can be download from the GitHub repository [1].

[1] https://github.com/dials/dials

Appendix A

Appendix

A.1 GLM background algorithm usage in *DIALS*

The command line parameters needed to invoke each method are listed in Table A.1. To set these parameters through *xia2*, they should be saved to a file (*e.g.* parameters.phil) and *xia2* called as follows:

```
# Call xia2 with DIALS specifying the integration parameters
xia2 -dials \
  dials.integrate.phil_file=parameters.phil \
  image=image_0001.cbf
```

A.2 Ice ring background algorithm usage in *DIALS*

The algorithm was implemented in C++ for use within *DIALS*. The global background model calculation is implemented as a separate command line program, *dials.model_background*. This program generates a file, *background.pickle*, which contains the computed global background model. It also generates a series of diagnostic images which can be used to inspect the properties of the dataset and the quality of the background model prior to integration. These include the minimum and maximum value at each pixel in the dataset and the mean, variance and index of dispersion (variance / mean) images. The mean image is used to generate the background model and the index of dispersion image is useful for evaluating the variation in the background at each pixel. Recalling that for a Poisson distribution, the index of dispersion, D = variance/mean = 1, then values significantly greater than 1.0 will indicate large variation in the counts for that pixel over the course of the dataset. An image of the final background model is also generated; viewing this allows a qualitative assessment of whether the generated model is appropriate for

Table A.1: The parameters required to invoke a particular background algorithm in *DIALS*.

Algorithm	Parameters
truncated	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=truncated`
nsigma	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=nsigma`
tukey	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=tukey`
plane	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=plane`
normal	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=normal`
null	`integration.background.algorithm=simple`
	`integration.background.simple.outlier.algorithm=null`
glm	`integration.background.algorithm=glm`

the data. The mean image or generated background model could also be used to provide automatic ice ring detection.

The background model is then applied in the *dials.integrate* program by setting the *background.algorithm=gmodel* user parameter to perform integration using the global background model algorithm. The robust or non-robust fitting algorithm can be selected *via* a user-parameter depending on what is most appropriate for the particular dataset. Currently, the input experiments file must contain the profile information generated from a successful integration run; therefore, an initial integration run is required before performing the global background modelling. In future versions of *DIALS*, these steps may be applied together. Sample program usage is shown below.

```
# Compute the global background model
dials.model_background integrated_experiment.json

# Integrate using the new global background model algorithm
dials.integrate \
  integrated_experiments.json \
  background.algorithm=gmodel \
  background.gmodel.robust.algorithm=True \
  background.gmodel.model=background.pickle
```

A.3 Ice ring background data processing, reduction and refinement

Aside from the choice of background model algorithm, the details of the processing were identical in each case. Each dataset was processed with *xia2* (Winter, 2009) using *DIALS* (Winter *et al.*, 2018) as the data analysis engine. The integrated *experiments.json* file produced by *xia2* after integration was then passed into a new program (dials.model_background) which was used to compute the global background model. The data was then integrated again, first using the default background algorithm and then using the global background model algorithm. In each case the data were integrated using summation integration and profile fitting. Reflections falling on ice rings were not excluded from the data processing. For datasets composed of more than one sweep, each sweep was integrated separately.

The data were processed using *xia2* with the *DIALS* pipeline as follows, where `${PATH_TO_IMAGES}` is a place holder for the path to the directory containing the image data.

```
# Run xia2 using the DIALS pipeline
xia2 pipeline=dials atom=X ${PATH_TO_IMAGES}
```

The procedure for re-integrating the data using the global background model algorithm and again with the default background algorithm is shown as follows where `${ORIGINAL_EXPERIMENTS}` is a place holder for the path to the integrated experiments.json file from the *xia2* processing.

```
# Compute the global background model
dials.model_background ${ORIGINAL_EXPERIMENTS}

# Integrate using the default background algorithm
dials.integrate \
  ${ORIGINAL_EXPERIMENTS} \
  background.algorithm=glm

# Integrate using the new global background model algorithm
dials.integrate \
  ${ORIGINAL_EXPERIMENTS} \
  background.algorithm=gmodel \
  background.gmodel.model=background.pickle
```

The data reduction was performed using *POINTLESS* (Evans, 2005), *AIMLESS* (Evans and Murshudov, 2013) and *CTRUNCATE* (Winn *et al.*, 2011) specifying the

known space group and the resolution as reported in the PDB entry for each dataset. For datasets composed of more than one sweep, the datasets were scaled and merged together in *AIMLESS* to produce a single merged MTZ file. A free set of reflections for cross-validation in refinement was then selected using the *FREERFLAG* program. The *UNIQUEIFY* script in the *CCP4i* GUI application (Winn *et al.*, 2011) was used to ensure that the same free set was used for the processing of all instances of the same PDB entry. Prior to refinement, the coordinates in the PDB file for the dataset were randomised using *PDBSET* (Winn *et al.*, 2011) with a maximum noise level of 0.4Å to ensure that there was no bias in the refinement and R_{free} calculation. Finally, each dataset was refined to convergence against the randomised structure using *REFMAC5* (Murshudov *et al.*, 2011).

A complete script detailing the data reduction and refinement steps is shown below where `${PDBID}`, `${SPACEGROUP}` and `${RESOLUTION}` are place holders for the known PDB identifier, space group and resolution respectively.

```
# Check the space group
pointless <<<EOF
HKLIN integrated.mtz
HKLOUT unscaled.mtz
CHOOSE SPACEGROUP ${SPACEGROUP}
END
EOF

# Scale the data
aimless <<<EOF
HKLIN unscaled.mtz
HKLOUT scaled.mtz
RESO HIGH ${RESOLUTION}
END
EOF

# Convert to amplitudes
ctruncate \
  -hklin scaled.mtz \
  -hklout truncated.mtz \
  -colin "/*/*/[IMEAN,SIGIMEAN]"

# Select the free set for validation
freerflag \
  HKLIN truncated.mtz \
```

```
  HKLOUT free.mtz

# Use CCP4i GUI to select same free set using "uniqueify"
# Then output file "unique.mtz" for each case.

# Randomise coordinates
pdbset \
  XYZIN ${PDBID}.pdb \
  XYZOUT ${PDBID}_randomised.pdb <<<EOF
NOISE 0.4
END
EOF

# Do the refinement
refmac5 \
  XYZIN ${PDBID}_randomised.pdb \
  XYZOUT refined.pdb \
  HKLIN unique.mtz \
  HKLOUT refined.mtz <<<EOF
MAKE NEWLIGAND CONTINUE -
  HYDROGEN ALL
MONI DISTANCE 1000000
NCYCLE 80
RIDGE DEST SIGMA 0.01
END
EOF
```

A.4 Block matrix inversion

If a matrix, $\boldsymbol{A}$, can be partitioned as follows:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_{11} & \boldsymbol{A}_{12} \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} \end{pmatrix}, \tag{A.1}$$

then, as shown in Petersen and Pedersen (2012), its inverse is

$$\boldsymbol{A}^{-1} = \begin{pmatrix} \boldsymbol{A}^{11} & \boldsymbol{A}^{12} \\ \boldsymbol{A}^{21} & \boldsymbol{A}^{22} \end{pmatrix}. \tag{A.2}$$

Where

$$
\begin{aligned}
\boldsymbol{A}^{11} &= (\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1} = \boldsymbol{A}_{11}^{-1} + \boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12}(\boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12})^{-1}\boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1} \\
\boldsymbol{A}^{12} &= -(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} = (\boldsymbol{A}^{21})^T \\
\boldsymbol{A}^{21} &= -\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1} = (\boldsymbol{A}^{12})^T \\
\boldsymbol{A}^{22} &= \boldsymbol{A}_{22}^{-1} + \boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21}(\boldsymbol{A}_{11} - \boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1}\boldsymbol{A}_{21})^{-1}\boldsymbol{A}_{12}\boldsymbol{A}_{22}^{-1} = (\boldsymbol{A}_{22} - \boldsymbol{A}_{21}\boldsymbol{A}_{11}^{-1}\boldsymbol{A}_{12})^{-1}.
\end{aligned}
\tag{A.3}
$$

A.5 Product of the Ewald sphere and RLP distributions

As shown in Equation 5.22, the joint distribution of the conditional reciprocal lattice distribution and the product of the marginal and Ewald sphere distributions then has a mean, $\boldsymbol{p}$, and covariance matrix, $\boldsymbol{P}$ give by:

$$
\begin{aligned}
\boldsymbol{p} &= \begin{pmatrix} \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{p}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{p}_2 \end{pmatrix} \\
\boldsymbol{P} &= \begin{pmatrix} \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(1-\kappa) & \kappa\boldsymbol{\Sigma}_{12} \\ \kappa\boldsymbol{\Sigma}_{21} & \kappa\boldsymbol{\Sigma}_{22} \end{pmatrix}.
\end{aligned}
\tag{A.4}
$$

This can be seen by doing the block matrix inversion of the covariance matrix:

$$
\boldsymbol{P}^{-1} = \begin{pmatrix} \boldsymbol{P}^{11} & \boldsymbol{P}^{12} \\ \boldsymbol{P}^{21} & \boldsymbol{P}^{22} \end{pmatrix}.
\tag{A.5}
$$

Where the components of the inverted matrix are

$$
\begin{aligned}
\boldsymbol{P}^{11} &= [\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(1-\kappa) - (\kappa\boldsymbol{\Sigma}_{12})(\kappa\boldsymbol{\Sigma}_{22})^{-1}(\kappa\boldsymbol{\Sigma}_{21})]^{-1} \\
&= [\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}]^{-1} \\
\boldsymbol{P}^{12} &= -[\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}]^{-1}(\kappa\boldsymbol{\Sigma}_{12})(\kappa\boldsymbol{\Sigma}_{22})^{-1} \\
&= -[\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}]^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} = (\boldsymbol{\Sigma}^{21})^T \\
\boldsymbol{P}^{22} &= (\kappa\boldsymbol{\Sigma}_{22})^{-1} + (\kappa\boldsymbol{\Sigma}_{22})^{-1}(\kappa\boldsymbol{\Sigma}_{21})[\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}]^{-1}(\kappa\boldsymbol{\Sigma}_{12})(\kappa\boldsymbol{\Sigma}_{22})^{-1} \\
&= (\kappa\boldsymbol{\Sigma}_{22})^{-1} + \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}[\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}]^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}.
\end{aligned}
\tag{A.6}
$$

Now $D^2 = (\boldsymbol{x} - \boldsymbol{p})^T\boldsymbol{P}^{-1}(\boldsymbol{x} - \boldsymbol{p})$ is expanded as follows:

$$
\begin{aligned}
D^2 &= (\boldsymbol{x}_1 - \boldsymbol{p}_1)^T \boldsymbol{P}^{11} (\boldsymbol{x}_1 - \boldsymbol{p}_1) + 2(\boldsymbol{x}_1 - \boldsymbol{p}_1)^T \boldsymbol{P}^{12} (\boldsymbol{x}_2 - \boldsymbol{p}_2) + (\boldsymbol{x}_2 - \boldsymbol{p}_2)^T \boldsymbol{P}^{22} (\boldsymbol{x}_2 - \boldsymbol{p}_2) \\
&= (\boldsymbol{x}_1 - \boldsymbol{p}_1)^T (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} (\boldsymbol{x}_1 - \boldsymbol{p}_1) \\
&\quad - 2(\boldsymbol{x}_1 - \boldsymbol{p}_1)^T (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2) \\
&\quad + (\boldsymbol{x}_2 - \boldsymbol{p}_2)^T [(\kappa\boldsymbol{\Sigma}_{22})^{-1} + \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}] (\boldsymbol{x}_2 - \boldsymbol{p}_2) \\
&= [(\boldsymbol{x}_1 - \boldsymbol{p}_1) - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2)]^T (\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21})^{-1} [(\boldsymbol{x}_1 - \boldsymbol{p}_1) - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2)] \\
&\quad + (\boldsymbol{x}_2 - \boldsymbol{p}_2)^T (\kappa\boldsymbol{\Sigma}_{22})^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2).
\end{aligned}
\tag{A.7}
$$

Now expanding for $\boldsymbol{p}_1$:

$$
\begin{aligned}
(\boldsymbol{x}_1 - \boldsymbol{p}_1) - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2) &= \boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{p}_2 - \boldsymbol{\mu}_2) - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2) \\
&= \boldsymbol{x}_1 - \boldsymbol{\mu}_1 - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_2) \\
&= \boldsymbol{x}_1 - \bar{\boldsymbol{\mu}}.
\end{aligned}
\tag{A.8}
$$

Which gives us the joint distribution of the conditional distribution and product of the marginal and Ewald sphere distributions:

$$
(\boldsymbol{x} - \boldsymbol{p})^T \boldsymbol{P}^{-1} (\boldsymbol{x} - \boldsymbol{p}) = (\boldsymbol{x}_1 - \bar{\boldsymbol{\mu}})^T \bar{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x}_1 - \bar{\boldsymbol{\mu}}) + (\boldsymbol{x}_2 - \boldsymbol{p}_2)^T (\kappa\boldsymbol{\Sigma}_{22})^{-1} (\boldsymbol{x}_2 - \boldsymbol{p}_2). \tag{A.9}
$$

A.6 Derivatives of the RLP parametrisation

The reciprocal lattice point covariance matrix is specified by parameters using the Cholesky decomposition. The covariance matrix, as shown in Equation 5.7 is

$$
\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^* = \begin{pmatrix} m_1^2 & m_2 m_1 & m_4 m_1 \\ m_2 m_1 & m_2^2 + m_3^2 & m_4 m_2 + m_5 m_3 \\ m_4 m_1 & m_4 m_2 + m_5 m_3 & m_4^2 + m_5^2 + m_6^2 \end{pmatrix}. \tag{A.10}
$$

The derivatives of the covariance matrix, $\boldsymbol{M}$, with respect to each of the parameters, $(m_1, m_2, m_3, m_4, m_5, m_6)$, is:

$$\frac{\partial \boldsymbol{M}}{\partial m_1} = \begin{pmatrix} 2m_1 & m_2 & m_4 \\ m_2 & 0 & 0 \\ m_4 & 0 & 0 \end{pmatrix}, \quad \frac{\partial \boldsymbol{M}}{\partial m_2} = \begin{pmatrix} 0 & m_1 & 0 \\ m_1 & 2m_2 & m_4 \\ 0 & m_4 & 0 \end{pmatrix}, \quad \frac{\partial \boldsymbol{M}}{\partial m_3} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2m_3 & m_5 \\ 0 & m_5 & 0 \end{pmatrix}$$
$$\frac{\partial \boldsymbol{M}}{\partial m_4} = \begin{pmatrix} 0 & 0 & m_1 \\ 0 & 0 & m_2 \\ m_1 & m_2 & 2m_4 \end{pmatrix}, \quad \frac{\partial \boldsymbol{M}}{\partial m_5} = \begin{pmatrix} 0 & 0 & m_1 \\ 0 & 0 & m_3 \\ m_1 & m_3 & 2m_5 \end{pmatrix}, \quad \frac{\partial \boldsymbol{M}}{\partial m_6} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2m_6 \end{pmatrix}. \tag{A.11}$$

A.7 Derivatives for δ-function wavelength model

For the monochromatic wavelength model, the derivatives of the quantities $\boldsymbol{\Sigma}_{XY}$, $\boldsymbol{\mu}_{XY}$, $\boldsymbol{\Sigma}_Z$ and ϵ with respect to the parameters, β, are as follows

$$\begin{aligned}
\frac{\partial \boldsymbol{\Sigma}_{XY}}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{11} - \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{12} \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{21} \\
\frac{\partial \boldsymbol{\mu}_{XY}}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_{1} + \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{12} \boldsymbol{\Sigma}_{22}^{-1}\epsilon - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} \boldsymbol{\Sigma}_{22}^{-1}\epsilon - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\frac{\partial \epsilon}{\partial \beta_k} \\
\frac{\partial \boldsymbol{\Sigma}_Z}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} \\
\frac{\partial \epsilon}{\partial \beta_k} &= -\left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_{2}.
\end{aligned} \tag{A.12}$$

For a given reflection, the rotation matrix, $\boldsymbol{R}_e$ transforms a point in reciprocal space into the local reflection specific coordinate system. The covariance matrix in the reflection specific coordinate system is then: $\boldsymbol{\Sigma} = \boldsymbol{R}_e \boldsymbol{M} {\boldsymbol{R}_e}^T$. Therefore:

$$\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k} = \boldsymbol{R}_e \left(\frac{\partial \boldsymbol{M}}{\partial \beta_k}\right) {\boldsymbol{R}_e}^T. \tag{A.13}$$

Where the derivatives of $\boldsymbol{M}$ with respect to the model parameters are given in Appendix A.6.

A.8 Derivatives for Normal wavelength model

For the Normal wavelength model, the derivatives of the quantities $\boldsymbol{\Sigma}_{XY}$, $\boldsymbol{\mu}_{XY}$, $\boldsymbol{\Sigma}_Z$ and ϵ with respect to the parameters, β, are as follows

$$
\begin{aligned}
\frac{\partial \boldsymbol{\Sigma}_{XY}}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{Q}}{\partial \beta_k}\right)_{11} \\
\frac{\partial \boldsymbol{\mu}_{XY}}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{q}}{\partial \beta_k}\right)_{1} \\
\frac{\partial \boldsymbol{\Sigma}_{Z}}{\partial \beta_k} &= \kappa \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} + \boldsymbol{\Sigma}_{22} \left(\frac{\partial \kappa}{\partial \beta_k}\right) \\
\frac{\partial \epsilon}{\partial \beta_k} &= -\left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_{2}.
\end{aligned}
\tag{A.14}
$$

The derivatives of the mean, $\boldsymbol{q}$, and covariance matrix, $\boldsymbol{Q}$, of the diffracted beam vectors are given by:

$$
\begin{aligned}
\frac{\partial \boldsymbol{q}}{\partial \beta_k} &= \left(\frac{1}{\boldsymbol{s}_0 \cdot \boldsymbol{p}}\right)\left[\left(\boldsymbol{p} \cdot \frac{\partial \boldsymbol{p}}{\partial \beta_k}\right) + \frac{1}{2}\left(\frac{\boldsymbol{p} \cdot \boldsymbol{p}}{\boldsymbol{s}_0 \cdot \boldsymbol{p}}\right)\left(\boldsymbol{s}_0 \cdot \frac{\partial \boldsymbol{p}}{\partial \beta_k}\right)\right]\boldsymbol{s}_0 + \frac{\partial \boldsymbol{p}}{\partial \beta_k} \\
\frac{\partial \boldsymbol{Q}}{\partial \beta_k} &= \left(\frac{\partial \boldsymbol{J}}{\partial \beta_k}\right)\boldsymbol{P}\boldsymbol{J}^T + \boldsymbol{J}\left(\frac{\partial \boldsymbol{P}}{\partial \beta_k}\right)\boldsymbol{J}^T + \boldsymbol{J}\boldsymbol{P}\left(\frac{\partial \boldsymbol{J}}{\partial \beta_k}\right)^T.
\end{aligned}
\tag{A.15}
$$

Where the derivatives of matrix, $\boldsymbol{J}$, are given by:

$$
\begin{aligned}
\frac{\partial \boldsymbol{J}}{\partial \beta_k} = -\frac{\boldsymbol{s}_0}{(\boldsymbol{p} \cdot \boldsymbol{s}_0)^2}\Bigg[&\left(\frac{\partial \boldsymbol{p}}{\partial \beta_k} \cdot \boldsymbol{s}_0\right)\boldsymbol{p} + (\boldsymbol{p} \cdot \boldsymbol{s}_0)\frac{\partial \boldsymbol{p}}{\partial \beta_k} - \left(\frac{\partial \boldsymbol{p}}{\partial \beta_k} \cdot \boldsymbol{p}\right)\boldsymbol{s}_0 \\
&- \frac{2(\boldsymbol{p} \cdot \boldsymbol{s}_0)\boldsymbol{p} - (\boldsymbol{p} \cdot \boldsymbol{p})\boldsymbol{s}_0}{\boldsymbol{p} \cdot \boldsymbol{s}_0}\left(\frac{\partial \boldsymbol{p}}{\partial \beta_k} \cdot \boldsymbol{s}_0\right)\Bigg].
\end{aligned}
\tag{A.16}
$$

The mean, $\boldsymbol{p}$, and covariance matrix, $\boldsymbol{P}$, of the product distribution of the Ewald sphere and RLP distributions can be partitioned as follows:

$$
\begin{aligned}
\boldsymbol{p} &= \begin{pmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \end{pmatrix} \\
\boldsymbol{P} &= \begin{pmatrix} \boldsymbol{P}_{11} & \boldsymbol{P}_{12} \\ \boldsymbol{P}_{12} & \boldsymbol{P}_{22} \end{pmatrix}.
\end{aligned}
\tag{A.17}
$$

The derivatives of the components of the mean, $\boldsymbol{p}$, are:

$$\left(\frac{\partial \boldsymbol{p}}{\partial \beta_k}\right)_1 = \left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_1 + \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{12} \boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{p}_2 - \boldsymbol{\mu}_2) - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} \boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{p}_2 - \boldsymbol{\mu}_2)$$
$$+ \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\left(\frac{\partial \boldsymbol{p}}{\partial \beta_k}\right)_2 - \left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_2\right)$$
$$\left(\frac{\partial \boldsymbol{p}}{\partial \beta_k}\right)_2 = \left(\frac{1}{\boldsymbol{\Sigma}_{22} + \sigma_E^2}\right)\left[\left(\frac{\partial \boldsymbol{\mu}}{\partial \beta_k}\right)_2 \sigma_E^2 + \boldsymbol{\mu}_2\left(\frac{\partial \sigma_E^2}{\partial \beta_k}\right) + |\boldsymbol{s}_0|\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} - \boldsymbol{p}_2\left(\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} + \left(\frac{\partial \sigma_E^2}{\partial \beta_k}\right)\right)\right]. \tag{A.18}$$

The derivatives of the components of the covariance matrix, $\boldsymbol{P}$, are:

$$\left(\frac{\partial \boldsymbol{P}}{\partial \beta_k}\right)_{11} = \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{11} - \left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{12} \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(1-\kappa) + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22} \boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(1-\kappa)$$
$$- \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{21}(1-\kappa) + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\left(\frac{\partial \kappa}{\partial \beta_k}\right)$$
$$\left(\frac{\partial \boldsymbol{P}}{\partial \beta_k}\right)_{12} = \left(\frac{\partial \kappa}{\partial \beta_k}\right)\boldsymbol{\Sigma}_{12} + \kappa\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{12}$$
$$\left(\frac{\partial \boldsymbol{P}}{\partial \beta_k}\right)_{21} = \left(\frac{\partial \kappa}{\partial \beta_k}\right)\boldsymbol{\Sigma}_{21} + \kappa\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{21}$$
$$\left(\frac{\partial \boldsymbol{P}}{\partial \beta_k}\right)_{22} = \left(\frac{\partial \kappa}{\partial \beta_k}\right)\boldsymbol{\Sigma}_{22} + \kappa\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22}. \tag{A.19}$$

The derivatives of the scale factor, κ, are:

$$\frac{\partial \kappa}{\partial \beta_k} = \left(\frac{1}{\boldsymbol{\Sigma}_{22} + \sigma_E^2}\right)\left((1-\kappa)\left(\frac{\partial \sigma_E^2}{\partial \beta_k}\right) - \kappa\left(\frac{\partial \boldsymbol{\Sigma}}{\partial \beta_k}\right)_{22}\right). \tag{A.20}$$

The derivatives of the Ewald sphere variance, σ_E^2, are:

$$\frac{\partial \sigma_E^2}{\partial \beta_k} = (\boldsymbol{r}_E \cdot \boldsymbol{r}_E)\left(\boldsymbol{r}_E \cdot \frac{\partial \boldsymbol{r}_E}{\partial \beta_k}\right)\sigma_\lambda^2 + \left(\frac{\boldsymbol{r}_E \cdot \boldsymbol{r}_E}{2}\right)^2 \frac{\partial \sigma_\lambda^2}{\partial \beta_k}. \tag{A.21}$$

The derivatives of the point on the Ewald sphere, $\boldsymbol{r}_E$, are:

$$\frac{\partial \boldsymbol{r}_E}{\partial \beta_k} = \frac{1}{|\boldsymbol{s}_2|}\left(\frac{\partial \boldsymbol{s}_2}{\partial \beta_k}\right) - \frac{\boldsymbol{s}_2}{|\boldsymbol{s}_2|^3}\left(\frac{\partial \boldsymbol{s}_2}{\partial \beta_k} \cdot \boldsymbol{s}_2\right). \tag{A.22}$$

A.9 Still image data processing

In order to process the data, the following procedure was performed. First the images were imported using the *dials.import* command. Then each image was processed independently using *dials.stills_process* as shown below.

```
# Import the data
dials.import ${PATH_TO_IMAGES}

# Run dials stills process
dials.stills_process \
  datablock.json \
  stills_process.phil
```

A joint refinement of the detector and beam models when then performed.

```
# Combine the experiments
dials.combine_experiments \
  *indexed.pickle
  *refined_experiments.json \
  reference_from_experiment.detector=0

# Perform a join refinement of the beam and detector
dials.refine \
  combined_experiments.json \
  combined_reflections.pickle \
  ../refine.phil
```

The data were then processed again using *dials.stills_process* now using the refined geometry; for this run, the detector and beam models remain fixed and only the unit cell and orientation are allowed to vary.

```
# Run dials stills process
dials.stills_process \
  datablock.json \
  stills_process_with_fixed_detector.phil
```

Finally, each image is processed independently with *dials.potato*. The input to the program is the set of strong spots and the refined experiments from *dials.stills_process* as follows:

```
# Process an image with potato
dials.potato \
  strong.pickle \
  refined_experiments.json
```

Bibliography

Abrahams, D. and R. W. Grosse-Kunstleve (2003). "Building hybrid systems with boost.python". In: *C/C++ Users Journal* 21.7, pp. 29–36. URL: https://www.osti.gov/biblio/815409.

Afonine, P. V., R. W. Grosse-Kunstleve, N. Echols, J. J. Headd, N. W. Moriarty, M. Mustyakimov, T. C. Terwilliger, A. Urzhumtsev, P. H. Zwart, and P. D. Adams (2012). "Towards automated crystallographic structure refinement with phenix.refine". In: *Acta Crystallographica Section D* 68.4, pp. 352–367. DOI: 10.1107/s0907444912001308.

Akaike, H. (1973). "Information theory and an extension of the maximum likelihood principle". In: *Proceedings of the 2nd international symposium on information theory*, pp. 267–281.

Alexander, L. E. and G. S. Smith (1962). "Single-crystal intensity measurements with the three-circle counter diffractometer". In: *Acta Crystallographica* 15.10, pp. 983–1004. DOI: 10.1107/s0365110x62002613.

Anscombe, F. J. (1948). "The transformation of Poisson, binomial and negative-binomial data". In: *Biometrika* 35.3-4, pp. 246–254. DOI: 10.1093/biomet/35.3-4.246.

Arndt, U. W. and A. J. Wonacott (1977). "Data collection from macromolecular crystals". In: *The rotation method in crystallography*. Ed. by U. W. Arndt and A. J. Wonacott. Vol. 1. 1. Amsterdam: North-Holland Publishing Company, p. 275. URL: https://books.google.co.uk/books?id=rDRCAQAAIAAJ.

Axford, D. (2018). personal communication.

Berman, H. M. (2000). "The Protein Data Bank". In: *Nucleic Acids Research* 28.1, pp. 235–242. DOI: 10.1093/nar/28.1.235.

Bernstein, H. J. and A. P. Hammersley (2006). "Specification of the crystallographic binary file (CBF/imgCIF)". In: *International tables for crystallography*. Ed. by S. R. Hall and B. McMahon. Chap. 2.3, pp. 37–43. DOI: 10.1107/97809553602060000729.

BioSync (2019). URL: http://biosync.sbkb.org/stats.do?stats_sec=MAIN&stats_focus_lvl=GLBL (visited on 2019-04-04).

Blanc, E., P. Roversi, C. Vonrhein, C. Flensburg, S. M. Lea, and G. Bricogne (2004). "Refinement of severely incomplete structures with maximum likelihood

in BUSTER - TNT". In: *Acta Crystallographica Section D* 60.12, pp. 2210–2221. DOI: 10.1107/s0907444904016427.

Bourgeois, D., D. Nurizzo, R. Kahn, and C. Cambillau (1998). "An integration routine based on profile fitting with optimized fitting area for the evaluation of weak and/or overlapped two-dimensional Laue or monochromatic patterns". In: *Journal of Applied Crystallography* 31.1, pp. 22–35. DOI: 10.1107/s0021889897006730.

Brewster, A. S., D. G. Waterman, J. M. Parkhurst, R. J. Gildea, T. M. Michels-Clark, I. D. Young, H. J. Bernstein, G. Winter, G. Evans, and N. K. Sauter (2016). "Processing XFEL data with cctbx.xfel and DIALS". In: *Computational Crystallography Newsletter* 7, pp. 32–53. URL: http://cci.lbl.gov/publications/download/CCN_2016_p32.pdf.

Brewster, A. S., D. G. Waterman, J. M. Parkhurst, R. J. Gildea, I. D. Young, L. J. O'Riordan, J. Yano, G. Winter, G. Evans, and N. K. Sauter (2018). "Improving signal strength in serial crystallography with DIALS geometry refinement". In: *Acta Crystallographica Section D* 74.9, pp. 877–894. DOI: 10.1107/s2059798318009191.

Bricogne, G. (1987). "The EEC Cooperative Programming Workshop on Position-Sensitive Detector Software". In: *Proceedings of the CCP4 Daresbury Study Weekend*, pp. 120–145. URL: http://purl.org/net/epubs/manifestation/943.

Bricogne, G. (1986). "Indexing and the Fourier transform". In: *Proceedings of the eec cooperative workshop on position-sensitive detector software (phase iii)*. Ed. by G. Bricogne, p. 28.

Bricogne, G., C. Vonrhein, C. Flensburg, M. Schiltz, and W. A. Paciorek (2003). "Generation, representation and flow of phase information in structure determination: recent developments in and around SHARP 2.0". In: *Acta Crystallographica Section D* 59.11, pp. 2023–2030. DOI: 10.1107/s0907444903017694.

Brünger, A. T. (1992). "Free R value: a novel statistical quantity for assessing the accuracy of crystal structures". In: *Nature* 355.6359, pp. 472–475. DOI: 10.1038/355472a0.

Cantoni, E. and E. Ronchetti (2001). "Robust inference for generalized linear models". In: *Journal of the American Statistical Association* 96.455, pp. 1022–1030. DOI: 10.1198/016214501753209004.

Casanas, A., R. Warshamanage, A. D. Finke, E. Panepucci, V. Olieric, A. Nöll, R. Tampé, S. Brandstetter, A. Förster, M. Mueller, C. Schulze-Briese, O. Bunk, and M. Wang (2016). "EIGER detector: application in macromolecular crystallography". In: *Acta Crystallographica Section D* 72.9, pp. 1036–1048. DOI: 10.1107/s2059798316012304.

Chapman, H. N., P. Fromme, A. Barty, T. A. White, R. A. Kirian, A. Aquila, M. S. Hunter, J. Schulz, D. P. DePonte, U. Weierstall, R. B. Doak, F. R. N. C. Maia, A. V. Martin, I. Schlichting, L. Lomb, N. Coppola, R. L. Shoeman, S. W. Epp,

R. Hartmann, D. Rolles, A. Rudenko, L. Foucar, N. Kimmel, G. Weidenspointner, P. Holl, M. Liang, M. Barthelmess, C. Caleman, S. Boutet, M. J. Bogan, J. Krzywinski, C. Bostedt, S. Bajt, L. Gumprecht, B. Rudek, B. Erk, C. Schmidt, A. Hömke, C. Reich, D. Pietschner, L. Strüder, G. Hauser, H. Gorke, J. Ullrich, S. Herrmann, G. Schaller, F. Schopper, H. Soltau, K.-U. Kühnel, M. Messerschmidt, J. D. Bozek, S. P. Hau-Riege, M. Frank, C. Y. Hampton, R. G. Sierra, D. Starodub, G. J. Williams, J. Hajdu, N. Timneanu, M. M. Seibert, J. Andreasson, A. Rocker, O. Jönsson, M. Svenda, S. Stern, K. Nass, R. Andritschke, C.-D. Schröter, F. Krasniqi, M. Bott, K. E. Schmidt, X. Wang, I. Grotjohann, J. M. Holton, T. R. M. Barends, R. Neutze, S. Marchesini, R. Fromme, S. Schorb, D. Rupp, M. Adolph, T. Gorkhover, I. Andersson, H. Hirsemann, G. Potdevin, H. Graafsma, B. Nilsson, and J. C. H. Spence (2011). "Femtosecond X-ray protein nanocrystallography". In: *Nature* 470.7332, pp. 73–77. DOI: 10.1038/nature09750.

Chapman, M. S. and T. Somasundaram (2010). "De-icing: recovery of diffraction intensities in the presence of ice rings". In: *Acta Crystallographica Section D* 66.6, pp. 741–744. DOI: 10.1107/s0907444910012436.

Choi, K. P. (1994). "On the medians of gamma distributions and an equation of Ramanujan". In: *Proceedings of the American Mathematical Society* 121.1, p. 245. DOI: 10.2307/2160389.

Cowtan, K. D. (2006). "The Buccaneer software for automated model building. 1. Tracing protein chains". In: *Acta Crystallographica Section D* 62.9, pp. 1002–1011. DOI: 10.1107/s0907444906022116.

Crockford, D. (2006). *The application/json media type for javascript object notation (JSON)*. Memo. IETF, pp. 1–11. URL: http://tools.ietf.org/html/rfc4627.txt.

Diamond, R. (1969). "Profile analysis in single-crystal diffractometry". In: *Acta Crystallographica Section A* 25.1, pp. 43–55. DOI: 10.1107/s0567739469000064.

Diederichs, K. (2015). personal communication.

Diederichs, K. (2019). *XDSGUI*. URL: https://strucbio.biologie.uni-konstanz.de/xdswiki/index.php/XDSGUI (visited on 2019-04-04).

Duisenberg, A. J. M., L. M. J. Kroon-Batenburg, and A. M. M. Schreurs (2003). "An intensity evaluation method: EVAL-14". In: *Journal of Applied Crystallography* 36.2, pp. 220–229. DOI: 10.1107/s0021889802022628.

Ebrahim, A., T. Moreno-Chicano, M. V. Appleby, A. K. Chaplin, J. H. Beale, D. A. Sherrell, H. M. E. Duyvesteyn, S. Owada, K. Tono, H. Sugimoto, R. W. Strange, J. A. R. Worrall, D. Axford, R. L. Owen, and M. A. Hough (2019). "Dose-resolved serial synchrotron and XFEL structures of radiation-sensitive metalloproteins". en. In: *IUCrJ* 6.4. DOI: 10.1107/S2052252519003956. (Visited on 2019-05-13).

Emsley, P., B. Lohkamp, W. G. Scott, and K. D. Cowtan (2010). "Features and development of Coot". In: *Acta Crystallographica Section D* 66.4, pp. 486–501. DOI: 10.1107/s0907444910007493.

Evans, G., D. Axford, and R. L. Owen (2011). "The design of macromolecular crystallography diffraction experiments". In: *Acta Crystallographica Section D* 67.4, pp. 261–270. DOI: 10.1107/s0907444911007608.

Evans, P. R. (2005). "Scaling and assessment of data quality". In: *Acta Crystallographica Section D* 62.1, pp. 72–82. DOI: 10.1107/s0907444905036693.

Evans, P. R. and A. J. McCoy (2007). "An introduction to molecular replacement". In: *Acta Crystallographica Section D* 64.1, pp. 1–10. DOI: 10.1107/s0907444907051554.

Evans, P. R. and G. N. Murshudov (2013). "How good are my data and what is the resolution?" In: *Acta Crystallographica Section D* 69.7, pp. 1204–1214. DOI: 10.1107/s0907444913000061.

Fisher, R. A. (1922). "On the mathematical foundations of theoretical statistics". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 222.594-604, pp. 309–368. DOI: 10.1098/rsta.1922.0009.

Foadi, J., P. Aller, Y. Alguel, A. Cameron, D. Axford, R. L. Owen, W. Armour, D. G. Waterman, S. Iwata, and G. Evans (2013). "Clustering procedures for the optimal selection of data sets from multiple crystals in macromolecular crystallography". In: *Acta Crystallographica Section D* 69.8, pp. 1617–1632. DOI: 10.1107/s0907444913012274.

Ford, G. C. (1974). "Intensity determination by profile-fitting applied to precession photographs". In: *Journal of Applied Crystallography* 7.6, pp. 555–564. DOI: 10.1107/s0021889874010430.

Ford, G. C. (1982). "The processing of oscillation photographs at sheffield". In: *Daresbury Laboratory Information Quarterly for Protein Crystallography* 9, pp. 3–7. URL: https://www.ccp4.ac.uk/newsletters/No09.pdf.

Ford, G. C. and J. S. Rollett (1968). "On versions of a procedure for scaling X-ray photographs". In: *Acta Crystallographica Section B* 24.2, pp. 293–294. DOI: 10.1107/s0567740868002190.

Fox, G. C. and K. C. Holmes (1966). "An alternative method of solving the layer scaling equations of Hamilton, Rollett and Sparks". In: *Acta Crystallographica* 20.6, pp. 886–891. DOI: 10.1107/s0365110x66002007.

French, S. and K. S. Wilson (1978). "On the treatment of negative intensity observations". In: *Acta Crystallographica Section A* 34.4, pp. 517–525. DOI: 10.1107/s0567739478001114.

French, S. and K. S. Wilson (2012). "Bayesian treatment of negative intensity measurements in crystallography". In: *Statistical Science*, pp. 1–6. URL: https:

//warwick.ac.uk/fac/sci/statistics/staff/academic-research/french/publications/bayes_impact_french_wilson.pdf.

Frome, E. L. (1982). "Algorithm AS 171: Fisher's exact variance test for the Poisson distribution". In: *Applied Statistics* 31.1, p. 67. DOI: 10.2307/2347079.

Fuentes-Landete, V., C. Mitterdorfer, P. H. Handle, G. N. Ruiz, S. Fuhrmann, and T. Loerting (2015). "Crystalline and amorphous ices". In: *Proceedings of the International School of Physics "Enrico Fermi"*, pp. 173–208. DOI: 10.3254/978-1-61499-507-4-173.

Fuentes-Montero, L. (2019). *DUI*. URL: https://github.com/ccp4/DUI (visited on 2019-04-10).

Gabanyi, M. J., P. D. Adams, K. Arnold, L. Bordoli, L. G. Carter, J. Flippen-Andersen, L. Gifford, J. Haas, A. Kouranov, W. A. McLaughlin, D. I. Micallef, W. Minor, R. Shah, T. Schwede, Y.-P. Tao, J. D. Westbrook, M. Zimmerman, and H. M. Berman (2011). "The structural biology knowledgebase: a portal to protein structures, sequences, functions, and methods". In: *Journal of Structural and Functional Genomics* 12.2, pp. 45–54. DOI: 10.1007/s10969-011-9106-2.

Gamma, E., R. Helm, R. Johnson, and J. Vlissides (1994). *Design patterns: elements of reusable object-oriented software*. 1st ed. Indianapolis: Addison-Wesley Professional.

Garman, E. F. (2010). "Radiation damage in macromolecular crystallography: what is it and why should we care?" en. In: *Acta Crystallographica Section D Biological Crystallography* 66.4, pp. 339–351. DOI: 10.1107/S0907444910008656. (Visited on 2019-07-30).

Garman, E. F. and R. L. Owen (2006). "Cryocooling and radiation damage in macromolecular crystallography". en. In: *Acta Crystallographica Section D Biological Crystallography* 62.1, pp. 32–47. DOI: 10.1107/S0907444905034207. (Visited on 2019-05-15).

Gati, C., G. P. Bourenkov, M. Klinge, D. Rehders, F. Stellato, D. Oberthür, O. Yefanov, B. P. Sommer, S. Mogk, M. Duszenko, C. Betzel, T. R. Schneider, H. N. Chapman, and L. Redecke (2014). "Serial crystallography on in vivo grown microcrystals using synchrotron radiation". In: *IUCrJ* 1.2, pp. 87–94. DOI: 10.1107/s2052252513033939.

Gildea, R. J., D. G. Waterman, J. M. Parkhurst, D. Axford, G. Sutton, D. I. Stuart, N. K. Sauter, G. Evans, and G. Winter (2014). "New methods for indexing multi-lattice diffraction data". In: *Acta Crystallographica Section D* 70.10, pp. 2652–2666. DOI: 10.1107/s1399004714017039.

Ginn, H. M., G. Evans, N. K. Sauter, and D. I. Stuart (2016). "On the release of cppxfel for processing X-ray free-electron laser images". In: *Journal of Applied Crystallography* 49.3, pp. 1065–1072. DOI: 10.1107/s1600576716006981.

Ginn, H. M., M. Messerschmidt, X. Ji, H. Zhang, D. Axford, R. J. Gildea, G. Winter, A. S. Brewster, J. Hattne, A. Wagner, J. M. Grimes, G. Evans, N. K. Sauter, G. Sutton, and D. I. Stuart (2015). "Structure of CPV17 polyhedrin determined by the improved analysis of serial femtosecond crystallographic data". In: *Nature Communications* 6.1, p. 6435. DOI: 10.1038/ncomms7435.

Greenhough, T. J. and F. L. Suddath (1986). "Oscillation camera data processing. 4. results and recommendations for the processing of synchrotron radiation data in macromolecular crystallography". In: *Journal of Applied Crystallography* 19.5, pp. 400–409. DOI: 10.1107/s002188988608915x.

Grosse-Kunstleve, R. W., N. K. Sauter, N. W. Moriarty, and P. D. Adams (2002). "The Computational Crystallography Toolbox: crystallographic algorithms in a reusable software framework". In: *Journal of Applied Crystallography* 35.1, pp. 126–136. DOI: 10.1107/s0021889801017824.

Guillemot, C. and O. Le Meur (2014). "Image Inpainting : Overview and Recent Advances". In: *IEEE Signal Processing Magazine* 31.1, pp. 127–144. DOI: 10.1109/MSP.2013.2273004.

Hall, J. P., K. O'Sullivan, A. Naseer, J. A. Smith, J. M. Kelly, and C. J. Cardin (2011). "Structure determination of an intercalating ruthenium dipyridophenazine complex which kinks DNA by semiintercalation of a tetraazaphenanthrene ligand". In: *Proceedings of the National Academy of Sciences* 108.43, pp. 17610–17614. DOI: 10.1073/pnas.1108685108.

Hamilton, W. C., J. S. Rollett, and R. A. Sparks (1965). "On the relative scaling of X-ray photographs". In: *Acta Crystallographica* 18.1, pp. 129–130. DOI: 10.1107/s0365110x65000233.

Harrison, S. C., F. K. Winkler, C. E. Schutt, and R. M. Durbin (1985). "Oscillation method with large unit cells". In: *Methods in enzymology*. Ed. by U. W. Arndt and A. J. Wonacott. Elsevier. Chap. 12, pp. 211–237. DOI: 10.1016/0076-6879(85)14021-8.

Hart, P., S. Boutet, G. Carini, M. Dubrovin, B. Duda, D. Fritz, G. Haller, R. Herbst, S. Herrmann, C. Kenney, N. Kurita, H. Lemke, M. Messerschmidt, M. Nordby, J. Pines, D. W. Schafer, M. Swift, M. Weaver, G. J. Williams, D. Zhu, N. Van Bakel, and J. Morse (2012). "The CSPAD megapixel X-ray camera at LCLS". In: *Proceedings of SPIE* 8504. Ed. by S. P. Moeller, M. Yabashi, and S. P. Hau-Riege, p. 85040C. DOI: 10.1117/12.930924.

Hasegawa, K., K. Yamashita, T. Murai, N. Nuemket, K. Hirata, G. Ueno, H. Ago, T. Nakatsu, T. Kumasaka, and M. Yamamoto (2017). "Development of a dose-limiting data collection strategy for serial synchrotron rotation crystallography". In: *Journal of Synchrotron Radiation* 24.1, pp. 29–41. DOI: 10.1107/s1600577516016362.

Hattne, J., N. Echols, R. Tran, J. Kern, R. J. Gildea, A. S. Brewster, R. Alonso-Mori, C. Glöckner, J. Hellmich, H. Laksmono, R. G. Sierra, B. Lassalle-Kaiser, A. Lampe, G. Han, S. Gul, D. DiFiore, D. Milathianaki, A. R. Fry, A. Miahnahri, W. E. White, D. W. Schafer, M. M. Seibert, J. E. Koglin, D. Sokaras, T.-C. Weng, J. Sellberg, M. J. Latimer, P. Glatzel, P. H. Zwart, R. W. Grosse-Kunstleve, M. J. Bogan, M. Messerschmidt, G. J. Williams, S. Boutet, J. Messinger, A. Zouni, J. Yano, U. Bergmann, V. K. Yachandra, P. D. Adams, and N. K. Sauter (2014). "Accurate macromolecular structures using minimal measurements from X-ray free-electron lasers". In: *Nature Methods* 11.5, pp. 545–548. DOI: 10.1038/nmeth.2887.

Hauptman, H. (1997). "Phasing methods for protein crystallography". In: *Current Opinion in Structural Biology* 7.5, pp. 672–680. DOI: 10.1016/s0959-440x(97)80077-2.

Helliwell, J. R. and E. P. Mitchell (2015). "Synchrotron radiation macromolecular crystallography: science and spin-offs". In: *IUCrJ* 2.2, pp. 283–291. DOI: 10.1107/s205225251402795x.

Henrich, B., A. Bergamaschi, C. Broennimann, R. Dinapoli, E. F. Eikenberry, I. Johnson, M. Kobas, P. Kraft, A. Mozzanica, and B. Schmitt (2009). "PILATUS: a single photon counting pixel detector for X-ray applications". In: *Nuclear Instruments and Methods in Physics Research Section A* 607.1, pp. 247–249. DOI: 10.1016/j.nima.2009.03.200.

Heritier, S., E. Cantoni, S. Copt, and M.-P. Victoria-Feser (2009). "Computations for the robust GEE estimator". In: *Robust methods in biostatistics*. Ed. by S. Heritier, E. Cantoni, S. Copt, and M.-P. Victoria-Feser. John Wiley and Sons. Chap. Appendix E, pp. 239–243. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470740538.

Herrmann, S., P. Hart, A. Dragone, D. Freytag, R. Herbst, J. Pines, M. Weaver, G. A. Carini, J. B. Thayer, O. Shawn, C. J. Kenney, and G. Haller (2014). "CSPAD upgrades and CSPAD v1.5 at LCLS". In: *Journal of Physics: Conference Series* 493, p. 012013.

Holton, J. (2018). *nanoBragg*. URL: https://bl831.als.lbl.gov/~jamesh/nanoBragg/ (visited on 2019-02-20).

Huber, P. J. (1964). "Robust estimation of a location parameter". In: *Annals of Mathematical Statistics* 35.1, pp. 73–101. DOI: 10.1214/aoms/1177703732.

James, R. W. (1948). *The optical principles of the diffraction of X-rays*. Ed. by W. L. Bragg. 1st ed. G. Bell and Sons. URL: https://doi.org/10.1016/0016-7037(66)90112-8.

Kabsch, W. (1988). "Evaluation of single-crystal X-ray diffraction data from a position-sensitive detector". In: *Journal of Applied Crystallography* 21.6, pp. 916–924. DOI: 10.1107/s0021889888007903.

Kabsch, W. (1993). "Recent extensions of the data processing program XDS". In: *Proceedings of the CCP4 study weekend*, pp. 63–70.

Kabsch, W. (2010a). "Integration, scaling, space-group assignment and post-refinement". In: *Acta Crystallographica Section D* 66.2, pp. 133–144. DOI: 10.1107/s0907444909047374.

Kabsch, W. (2010b). "XDS". In: *Acta Crystallographica Section D* 66.2, pp. 125–132. DOI: 10.1107/s0907444909047337.

Kabsch, W. (2014). "Processing of X-ray snapshots from crystals in random orientations". In: *Acta Crystallographica Section D* 70.8, pp. 2204–2216. DOI: 10.1107/s1399004714013534.

Knudsen, E. B., H. O. Sørensen, J. P. Wright, G. Goret, and J. Kieffer (2013). "FabIO: easy access to two-dimensional X-ray detector images in python". In: *Journal of Applied Crystallography* 46.2, pp. 537–539. DOI: 10.1107/s0021889813000150.

Kroon-Batenburg, L. M. J., A. M. M. Schreurs, R. B. G. Ravelli, and P. Gros (2015). "Accounting for partiality in serial crystallography using ray-tracing principles". In: *Acta Crystallographica Section D* 71.9, pp. 1799–1811. DOI: 10.1107/s1399004715011803.

Langer, G., S. X. Cohen, V. S. Lamzin, and A. Perrakis (2008). "Automated macromolecular model building for X-ray crystallography using ARP/wARP version 7". In: *Nature Protocols* 3.7, pp. 1171–1179. DOI: 10.1038/nprot.2008.91.

Lehmann, M. S. and F. K. Larsen (1974). "A method for location of the peaks in step-scan measured Bragg reflexions". In: *Acta Crystallographica Section A* 30.4, pp. 580–584. DOI: 10.1107/s0567739474001379.

Leslie, A. G. W. (1987). "Profile fitting". In: *Proceedings of the CCP4 study weekend*, pp. 39–50.

Leslie, A. G. W. (1999). "Integration of macromolecular diffraction data". In: *Acta Crystallographica Section D* 55.10, pp. 1696–1702. DOI: 10.1107/s090744499900846x.

Leslie, A. G. W. (2005). "The integration of macromolecular diffraction data". In: *Acta Crystallographica Section D* 62.1, pp. 48–57. DOI: 10.1107/s0907444905039107.

Leslie, A. G. W. and H. R. Powell (2007). "Processing diffraction data with MOSFLM". In: *Evolving methods for macromolecular crystallography*. Ed. by R. J. Read and J. L. Sussman. Springer Netherlands, pp. 41–51. DOI: 10.1007/978-1-4020-6316-9_4.

Leslie, A. G. W., H. R. Powell, G. Winter, O. Svensson, D. Spruce, S. McSweeney, D. Love, S. Kinder, E. Duke, and C. Nave (2002). "Automation of the collection and processing of X-ray diffraction data – a generic approach". In: *Acta Crystallographica Section D* 58.11, pp. 1924–1928. DOI: 10.1107/s0907444902016864.

Long, F., R. A. Nicholls, P. Emsley, S. Gražulis, A. Merkys, A. Vaitkus, and G. N. Murshudov (2017). "AceDRG : a stereochemical description generator for ligands". en. In: *Acta Crystallographica Section D Structural Biology* 73.2, pp. 112–122. DOI: 10.1107/S2059798317000067. (Visited on 2019-05-02).

McCoy, A. J., R. W. Grosse-Kunstleve, P. D. Adams, M. D. Winn, L. C. Storoni, and R. J. Read (2007). "Phaser crystallographic software". In: *Journal of Applied Crystallography* 40.4, pp. 658–674. DOI: 10.1107/s0021889807021206.

Messerschmidt, A. and J. W. Pflugrath (1987). "Crystal orientation and X-ray pattern prediction routines for area-detector diffractometer systems in macromolecular crystallography". In: *Journal of Applied Crystallography* 20.4, pp. 306–315. DOI: 10.1107/s002188988708662x.

Mitchell, E. P. and E. F. Garman (1994). "Flash freezing of protein crystals: investigation of mosaic spread and diffraction limit with variation of cryoprotectant concentration". In: *Journal of Applied Crystallography* 27.6, pp. 1069–1074. DOI: 10.1107/s0021889894008629.

Morris, R. J. and G. Bricogne (2003). "Sheldrick's 1.2 Å rule and beyond". In: *Acta Crystallographica Section D* 59.3, pp. 615–617. DOI: 10.1107/s090744490300163x.

Mozzanica, A., M. Andrä, R. Barten, A. Bergamaschi, S. Chiriotti, M. Brückner, R. Dinapoli, E. Fröjdh, D. Greiffenberg, F. Leonarski, C. Lopez-Cuenca, D. Mezza, S. Redford, C. Ruder, B. Schmitt, X. Shi, D. Thattil, G. Tinti, S. Vetter, and J. Zhang (2018). "The JUNGFRAU Detector for Applications at Synchrotron Light Sources and XFELs". In: *Synchrotron Radiation News* 31.6, pp. 16–20. DOI: 10.1080/08940886.2018.1528429.

Mueller, C., A. Marx, S. W. Epp, Y. Zhong, A. Kuo, A. R. Balo, J. Soman, F. Schotte, H. T. Lemke, R. L. Owen, E. F. Pai, A. R. Pearson, J. S. Olson, P. A. Anfinrud, O. P. Ernst, and R. J. Dwayne Miller (2015). "Fixed target matrix for femtosecond time-resolved and in situ serial micro-crystallography". In: *Structural Dynamics* 2.5, p. 054302. DOI: 10.1063/1.4928706.

Mueller, M., M. Wang, and C. Schulze-Briese (2011). "Optimal fine φ-slicing for single-photon-counting pixel detectors". In: *Acta Crystallographica Section D* 68.1, pp. 42–56. DOI: 10.1107/s0907444911049833.

Murshudov, G. N., P. Skubák, A. A. Lebedev, N. S. Pannu, R. A. Steiner, R. A. Nicholls, M. D. Winn, F. Long, and A. A. Vagin (2011). "REFMAC5 for the refinement of macromolecular crystal structures". In: *Acta Crystallographica Section D* 67.4, pp. 355–367. DOI: 10.1107/s0907444911001314.

Nakane, T. (2018). personal communication.

Nave, C. (1989). "Protein crystallography on a synchrotron". In: *Synchrotron Radiation News* 2.3, pp. 24–28. DOI: 10.1080/08940888908261212.

Nave, C. (1998). "A description of imperfections in protein crystals". In: *Acta Crystallographica Section D* 54.5, pp. 848–853. DOI: 10.1107/s0907444998001875.

Nelder, J. A. and R. W. M. Wedderburn (1972). "Generalized linear models". In: *Journal of the Royal Statistical Society. Series A (General)* 135.3, p. 370. DOI: 10.2307/2344614.

Neutze, R., R. Wouts, D. van der Spoel, E. Weckert, and J. Hajdu (2000). "Potential for biomolecular imaging with femtosecond X-ray pulses". en. In: *Nature* 406.6797, pp. 752–757. DOI: 10.1038/35021099. (Visited on 2019-05-02).

NIST (2004). *X-Ray Mass Attenuation Coefficients*.

Nyborg, J. and A. J. Wonacott (1977). "Computer programs". In: *The rotation method in crystallography*. Ed. by U. W. Arndt and A. J. Wonacott. Vol. 1. 1. Amsterdam: North-Holland Publishing Company, pp. 131–141. URL: https://books.google.co.uk/books?id=rDRCAQAAIAAJ.

Nyborg, J., A. J. Wonacott, J. C. Thierry, and J. N. Champness (1975). *Program IDXREF*.

Osborne, M. R. (1992). "Fisher's Method of Scoring". en. In: *International Statistical Review / Revue Internationale de Statistique* 60.1, p. 99. DOI: 10.2307/1403504. (Visited on 2019-05-16).

Otwinowski, Z., D. Borek, W. Majewski, and W. Minor (2003). "Multiparametric scaling of diffraction intensities". In: *Acta Crystallographica Section A* 59.3, pp. 228–234. DOI: 10.1107/s0108767303005488.

Otwinowski, Z. and W. Minor (1997). "Processing of X-ray diffraction data collected in oscillation mode". In: *Methods in Enzymology* 276, pp. 307–326. DOI: 10.1016/S0076-6879(97)76066-X.

Otwinowski, Z., W. Minor, D. Borek, and M. Cymborowski (2012). "DENZO and SCALEPACK". In: *International tables for crystallography volume f: crystallography of biological macromolecules*. Ed. by E. Arnold, D. M. Himmel, and M. G. Rossmann. Chap. 11.4, pp. 282–295. DOI: 10.1107/9780955360206000833.

Owen, R. L., D. Axford, D. A. Sherrell, A. Kuo, O. P. Ernst, E. C. Schulz, R. J. D. Miller, and H. M. Mueller-Werkmeister (2017). "Low-dose fixed-target serial synchrotron crystallography". In: *Acta Crystallographica Section D* 73.4, pp. 373–378. DOI: 10.1107/s2059798317002996.

Owen, R. L., J. Juanhuix, and M. Fuchs (2016). "Current advances in synchrotron radiation instrumentation for MX experiments". In: *Archives of Biochemistry and Biophysics* 602, pp. 21–31. DOI: 10.1016/j.abb.2016.03.021.

Padilla, J. E. and T. O. Yeates (2003). "A statistic for local intensity differences: robustness to anisotropy and pseudo-centering and utility for detecting twinning". In: *Acta Crystallographica Section D* 59.7, pp. 1124–1130. DOI: 10.1107/s0907444903007947.

Parkhurst, J. M., A. S. Brewster, L. Fuentes-Montero, D. G. Waterman, J. Hattne, A. W. Ashton, N. Echols, G. Evans, N. K. Sauter, and G. Winter (2014). "DXTBX: the diffraction experiment toolbox". In: *Journal of Applied Crystallography* 47.4, pp. 1459–1465. DOI: 10.1107/s1600576714011996.

Parkhurst, J. M., A. R. S. Thorn, M. Vollmar, G. Winter, D. G. Waterman, L. Fuentes-Montero, R. J. Gildea, G. N. Murshudov, and G. Evans (2017). "Background modelling of diffraction data in the presence of ice rings". In: *IUCrJ* 4.5, pp. 626–638. DOI: 10.1107/s2052252517010259.

Parkhurst, J. M., G. Winter, D. G. Waterman, L. Fuentes-Montero, R. J. Gildea, G. N. Murshudov, and G. Evans (2016). "Robust background modelling in DIALS". In: *Journal of Applied Crystallography* 49.6, pp. 1912–1921. DOI: 10.1107/s1600576716013595.

Patterson, A. L. (1934). "A Fourier series method for the determination of the components of interatomic distances in crystals". In: *Physical Review* 46.5, pp. 372–376. DOI: 10.1103/physrev.46.372.

PDB (2019a). *PDB Data Distribution by Resolution*. URL: https://www.rcsb.org/stats/distribution_resolution (visited on 2019-04-03).

PDB (2019b). *PDB Statistics: Data Distribution by Experimental Method and Molecular Type*. URL: https://www.rcsb.org/stats/summary (visited on 2019-04-03).

PDB (2019c). *PDB Statistics: Growth of Structures from X-ray Crystallography Experiments Released per Year*. URL: https://www.rcsb.org/stats/growth/xray (visited on 2019-04-03).

Petersen, K. B. and M. S. Pedersen (2012). *The Matrix Cookbook*. en. 2012-11-15. Technical University of Denmark. URL: https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf.

Pflugrath, J. W. (1997). "Diffraction-data processing for electronic detectors: theory and practice". In: *Methods in enzymology*. Ed. by C. W. Carter. Vol. 276. 269. Elsevier, pp. 286–306. DOI: 10.1016/s0076-6879(97)76065-8.

Pflugrath, J. W. (1999). "The finer things in X-ray diffraction data collection". In: *Acta Crystallographica Section D* 55.10, pp. 1718–1725. DOI: 10.1107/s090744499900935x.

Phillips, J. C., A. Wlodawer, M. M. Yevitz, and K. O. Hodgson (1976). "Applications of synchrotron radiation to protein crystallography: preliminary results." In: *Proceedings of the National Academy of Sciences* 73.1, pp. 128–132. DOI: 10.1073/pnas.73.1.128.

Polyviou, D., M. M. Machelett, A. Hitchcock, A. J. Baylay, F. Macmillan, C. M. Moore, T. S. Bibby, and I. Tews (2018). "Structural and functional characterization of IdiA/FutA (Tery _ 3377), an iron-binding protein from the ocean

diazotroph Trichodesmium erythraeum". In: *Journel of Biological Chemistry* 293.18, pp. 18099–18109.

Porta, J., J. J. Lovelace, A. M. M. Schreurs, L. M. J. Kroon-Batenburg, and G. E. O. Borgstahl (2011). "Processing incommensurately modulated protein diffraction data with *Eval* 15". en. In: *Acta Crystallographica Section D Biological Crystallography* 67.7, pp. 628–638. DOI: 10.1107/S0907444911017884. (Visited on 2019-05-14).

Powell, H. R. (1999). "The Rossmann Fourier autoindexing algorithm in MOSFLM". In: *Acta Crystallographica Section D* 55.10, pp. 1690–1695. DOI: 10.1107/s0907444999009506.

Powell, H. R., T. G. G. Battye, L. Kontogiannis, O. Johnson, and A. G. W. Leslie (2017). "Integrating macromolecular X-ray diffraction data with the graphical user interface iMOSFLM". In: *Nature Protocols* 12.7, pp. 1310–1325. DOI: 10.1038/nprot.2017.037.

Remacle, F. and G. Winter (2007). "Diffraction image: a new CCP4 library". In: *CCP4 newsletter on protein crystallography*. URL: https://www.ccp4.ac.uk/newsletters/newsletter45/articles/DiffractionImage.pdf.

Rossmann, M. G. (1972). *The molecular replacement method. a collection of papers on the use of non-crystallographic symmetry*. Gordon and Breach. DOI: 10.1002/crat.19730081215.

Rossmann, M. G. and C. G. van Beek (1999). "Data processing". In: *Acta Crystallographica Section D* 55.10, pp. 1631–1640. DOI: 10.1107/s0907444999008379.

Rossmann, M. G., A. G. W. Leslie, S. S. Abdel-Meguid, and T. Tsukihara (1979). "Processing and post-refinement of oscillation camera data". In: *Journal of Applied Crystallography* 12.6, pp. 570–581. DOI: 10.1107/s0021889879013273.

Sauter, N. K. (2015). "XFEL diffraction: developing processing methods to optimize data quality". In: *Journal of Synchrotron Radiation* 22.2, pp. 239–248. DOI: 10.1107/s1600577514028203.

Sauter, N. K., R. W. Grosse-Kunstleve, and P. D. Adams (2004). "Robust indexing for automatic data collection". In: *Journal of Applied Crystallography* 37.3, pp. 399–409. DOI: 10.1107/s0021889804005874.

Sauter, N. K., J. Hattne, A. S. Brewster, N. Echols, P. H. Zwart, and P. D. Adams (2014). "Improved crystal orientation and physical properties from single-shot XFEL stills". In: *Acta Crystallographica Section D* 70.12, pp. 3299–3309. DOI: 10.1107/s1399004714024134.

Sauter, N. K., J. Hattne, R. W. Grosse-Kunstleve, and N. Echols (2013). "New python-based methods for data processing". In: *Acta Crystallographica Section D* 69.7, pp. 1274–1282. DOI: 10.1107/s0907444913000863.

Schreurs, A. M. M., X. Xian, and L. M. J. Kroon-Batenburg (2009). "EVAL15: aA diffraction data integration method based onab initiopredicted profiles". In: *Journal of Applied Crystallography* 43.1, pp. 70–82. DOI: 10.1107/s0021889809043234.

Schwager, P., K. S. Bartels, and A. Jones (1975). "Refinement of setting angles in screenless film methods". In: *Journal of Applied Crystallography* 8.2, pp. 275–280. DOI: 10.1107/s002188987501045x.

Sheldrick, G. M. (1990). "Phase annealing in SHELX-90: direct methods for larger structures". In: *Acta Crystallographica Section A* 46.6, pp. 467–473. DOI: 10.1107/s0108767390000277.

Sheldrick, G. M. (2007). "A short history of SHELX". In: *Acta Crystallographica Section A* 64.1, pp. 112–122. DOI: 10.1107/s0108767307043930.

Sheldrick, G. M. (2015). "Crystal structure refinement with SHELXL". In: *Acta Crystallographica Section C* 71.1, pp. 3–8. DOI: 10.1107/s2053229614024218.

Sherrell, D. A., A. J. Foster, L. Hudson, B. Nutter, J. O'Hea, S. Nelson, O. Paré-Labrosse, S. Oghbaey, R. J. D. Miller, and R. L. Owen (2015). "A modular and compact portable mini-endstation for high-precision, high-speed fixed target serial crystallography at FEL and synchrotron sources". In: *Journal of Synchrotron Radiation* 22.6, pp. 1372–1378. DOI: 10.1107/s1600577515016938.

Stein, N. (2007). "Why the moments of E take the values they do". In: *CCP4 newsletter on protein crystallography* 47, pp. 2–5. URL: http://www.ccp4.ac.uk/newsletters/newsletter47/newsletter47.pdf.

Stellato, F., D. Oberthür, M. Liang, R. Bean, C. Gati, O. Yefanov, A. Barty, A. Burkhardt, P. Fischer, L. Galli, R. A. Kirian, J. Meyer, S. Panneerselvam, C. H. Yoon, F. Chervinskii, E. Speller, T. A. White, C. Betzel, A. Meents, and H. N. Chapman (2014). "Room-temperature macromolecular serial crystallography using synchrotron radiation". In: *IUCrJ* 1.4, pp. 204–212. DOI: 10.1107/s2052252514010070.

Steller, I., R. Bolotovsky, and M. G. Rossmann (1997). "An algorithm for automatic indexing of oscillation images using Fourier analysis". In: *Journal of Applied Crystallography* 30.6, pp. 1036–1040. DOI: 10.1107/s0021889897008777.

Sutherland, I. E. and G. W. Hodgman (1974). "Reentrant polygon clipping". In: *Communications of the ACM* 17.1, pp. 32–42. DOI: 10.1145/360767.360802.

Terwilliger, T. C. (2003). "Improving macromolecular atomic models at moderate resolution by automated iterative model building, statistical density modification and refinement". In: *Acta Crystallographica Section D* 59.7, pp. 1174–1182. DOI: 10.1107/s0907444903009922.

Terwilliger, T. C., G. Bunkóczi, L.-W. Hung, P. H. Zwart, J. L. Smith, D. L. Akey, and P. D. Adams (2016). "Can I solve my structure by SAD phasing? anomalous

signal in SAD phasing". In: *Acta Crystallographica Section D* 72.3, pp. 346–358. DOI: 10.1107/s2059798315019269.

The HDF Group (1997). *Hierarchical Data Format, version 5*. URL: http://www.hdfgroup.org/HDF5/ (visited on 2019-04-12).

Thomas, D. J. (1992). "Modern equations of diffractometry. Diffraction geometry". In: *Acta Crystallographica Section A* 48.2, pp. 134–158. DOI: 10.1107/s0108767391008577.

Thompson, M. C., D. Cascio, and T. O. Yeates (2018). "Microfocus diffraction from different regions of a protein crystal: structural variations and unit-cell polymorphism". In: *Acta Crystallographica Section D* 74.5, pp. 411–421. DOI: 10.1107/s2059798318003479.

Thorn, A. R. S., J. M. Parkhurst, P. Emsley, R. A. Nicholls, M. Vollmar, G. Evans, and G. N. Murshudov (2017). "AUSPEX: a graphical tool for X-ray diffraction data analysis". In: *Acta Crystallographica Section D* 73.9, pp. 729–737. DOI: 10.1107/s205979831700969x.

Uervirojnangkoorn, M., O. B. Zeldin, A. Y. Lyubimov, J. Hattne, A. S. Brewster, N. K. Sauter, A. T. Brünger, and W. I. Weis (2015). "Enabling X-ray free electron laser crystallography for challenging biological systems from a limited number of crystals". In: *eLife* 4.4, pp. 1–29. DOI: 10.7554/elife.05421.

Vagin, A. A. and A. Teplyakov (1997). "MOLREP: an automated program for molecular replacement". In: *Journal of Applied Crystallography* 30.6, pp. 1022–1025. DOI: 10.1107/s0021889897006766.

Wagner, A., R. Duman, K. Henderson, and V. Mykhaylyk (2016). "In-vacuum long-wavelength macromolecular crystallography". In: *Acta Crystallographica Section D* 72.3, pp. 430–439. DOI: 10.1107/s2059798316001078.

Waldrop, M. M. (2016). "The chips are down for Moore's law". In: *Nature* 530.7589, pp. 144–147. DOI: 10.1038/530144a.

Waterman, D. G., G. Winter, R. J. Gildea, J. M. Parkhurst, A. S. Brewster, N. K. Sauter, and G. Evans (2016). "Diffraction-geometry refinement in the DIALS framework". In: *Acta Crystallographica Section D* 72.4, pp. 558–575. DOI: 10.1107/s2059798316002187.

Waterman, D. G., G. Winter, J. M. Parkhurst, L. Fuentes-Montero, J. Hattne, A. S. Brewster, N. K. Sauter, and G. Evans (2013). "The DIALS framework for integration software". In: *CCP4 newsletter on protein crystallography* 49, pp. 16–19. URL: http://www.ccp4.ac.uk/newsletters/newsletter49/articles/DIALS.pdf.

Weierstall, U., D. James, C. Wang, T. A. White, D. Wang, W. Liu, J. C. H. Spence, R. Bruce Doak, G. Nelson, P. Fromme, R. Fromme, I. Grotjohann, C. Kupitz, N. A. Zatsepin, H. Liu, S. Basu, D. Wacker, G. Won Han, V. Katritch, S. Boutet, M. Messerschmidt, G. J. Williams, J. E. Koglin, M. Marvin Seibert,

M. Klinker, C. Gati, R. L. Shoeman, A. Barty, H. N. Chapman, R. A. Kirian, K. R. Beyerlein, R. C. Stevens, D. Li, S. T. A. Shah, N. Howe, M. Caffrey, and V. Cherezov (2014). "Lipidic cubic phase injector facilitates membrane protein serial femtosecond crystallography". en. In: *Nature Communications* 5.1, p. 3309. DOI: 10.1038/ncomms4309. (Visited on 2019-05-20).

Weinert, T., N. Olieric, R. Cheng, S. Brünle, D. James, D. Ozerov, D. Gashi, L. Vera, M. Marsh, K. Jaeger, F. Dworkowski, E. Panepucci, S. Basu, P. Skopintsev, A. S. Doré, T. Geng, R. M. Cooke, M. Liang, A. E. Prota, V. Panneels, P. Nogly, U. Ermler, G. Schertler, M. Hennig, M. O. Steinmetz, M. Wang, and J. Standfuss (2017). "Serial millisecond crystallography for routine room-temperature structure determination at synchrotrons". In: *Nature Communications* 8.1. DOI: 10.1038/s41467-017-00630-4.

White, T. A., V. Mariani, W. Brehm, O. Yefanov, A. Barty, K. R. Beyerlein, F. Chervinskii, L. Galli, C. Gati, T. Nakane, A. Tolstikova, K. Yamashita, C. H. Yoon, K. Diederichs, and H. N. Chapman (2016). "Recent developments in CrystFEL". In: *Journal of Applied Crystallography* 49.2, pp. 680–689. DOI: 10.1107/s1600576716004751.

Wierman, J. L., O. Paré-Labrosse, A. Sarracini, J. E. Besaw, M. J. Cook, S. Oghbaey, H. Daoud, P. Mehrabi, I. Kriksunov, A. Kuo, D. J. Schuller, S. Smith, O. P. Ernst, D. M. E. Szebenyi, S. M. Gruner, R. J. D. Miller, and A. D. Finke (2019). "Fixed-target serial oscillation crystallography at room temperature". en. In: *IUCrJ* 6.2, pp. 305–316. DOI: 10.1107/S2052252519001453. (Visited on 2019-05-20).

Wilson, A. J. C. (1949). "The probability distribution of X-ray intensities". In: *Acta Crystallographica* 2.5, pp. 318–321. DOI: 10.1107/s0365110x49000813.

Winkler, F. K., C. E. Schutt, and S. C. Harrison (1979). "The oscillation method for crystals with very large unit cells". In: *Acta Crystallographica Section A* 35.6, pp. 901–911. DOI: 10.1107/s0567739479002035.

Winn, M. D., C. C. Ballard, K. D. Cowtan, E. J. Dodson, P. Emsley, P. R. Evans, R. M. Keegan, E. B. Krissinel, A. G. W. Leslie, A. J. McCoy, S. J. McNicholas, G. N. Murshudov, N. S. Pannu, E. A. Potterton, H. R. Powell, R. J. Read, A. A. Vagin, and K. S. Wilson (2011). "Overview of the CCP4 suite and current developments". In: *Acta Crystallographica Section D* 67.4, pp. 235–242. DOI: 10.1107/s0907444910045749.

Winter, G. (2009). "xia2: an expert system for macromolecular crystallography data reduction". In: *Journal of Applied Crystallography* 43.1, pp. 186–190. DOI: 10.1107/s0021889809045701.

Winter, G. (2018). personal communication.

Winter, G. and D. Hall (2014). *Thaumatin / Diamond Light Source I04 user training.* DOI: 10.5281/zenodo.10271.

Winter, G. and J. P. Hall (2016). *Data from complex cation Λ-[Ru(1,4,5,8-tetraazaphe nanthrene)2(dipyridophenazine)]2+ with the oligonucleotide d(TCGGCGCCGA) recorded as part of ongoing research*. DOI: 10.5281/zenodo.49675.

Winter, G. and K. McAuley (2016). *Low dose, high multiplicity thermolysin X-ray diffraction data from Diamond Light Source beamline I03*. DOI: 10.5281/zenodo.49559.

Winter, G., D. G. Waterman, J. M. Parkhurst, A. S. Brewster, R. J. Gildea, M. Gerstel, L. Fuentes-Montero, M. Vollmar, T. M. Michels-Clark, I. D. Young, N. K. Sauter, and G. Evans (2018). "DIALS: implementation and evaluation of a new integration package". In: *Acta Crystallographica Section D* 74.2, pp. 85–97. DOI: 10.1107/s2059798317017235.

Wojdyr, M. (2019). *Gemmi: PDB stats*. URL: https://project-gemmi.github.io/pdb-stats/xray.html (visited on 2019-05-02).

Yeates, T. O. (1997). "Detecting and overcoming crystal twinning". In: *Methods in enzymology*. Ed. by C. W. Carter. Elsevier. Chap. 22, pp. 344–358. DOI: 10.1016/s0076-6879(97)76068-3.

Zachariasen, W. H. (1945). *Theory of X-ray diffraction in crystals*. 1st ed. John Wiley and Sons.

Zwart, P. H., R. W. Grosse-Kunstleve, and P. D. Adams (2005). "Characterization of X-ray data sets". In: *CCP4 newsletter on protein crystallography*. URL: https://www.phenix-online.org/papers/ccp4_july_2005_zwart.pdf.